# REPORT DOCUMENTATION PAGE

*Form Approved*
*OMB No. 0704-0188*

| | | |
|---|---|---|
| **1. AGENCY USE ONLY** *(Leave blank)* | **2. REPORT DATE**<br>August 1986 | **3. REPORT TYPE AND DATES COVERED**<br>Technical |

**4. TITLE AND SUBTITLE**
Computer Program Development Specification For The Surface Wave Analysis System (SWANS)

**5. FUNDING NUMBERS**

**6. AUTHORS**
K.D. Hutchenson, R.s. Cunfer, and A.F. Sutton

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**
ENSCO, Inc.
4849 N. Wickham Rd.
Melbourne, FL 32940

**8. PERFORMING ORGANIZATION REPORT NUMBER**
DCS-SFS-86-02

**9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)**
Air Force Technical Applications Center (AFTAC/TTR)
1030 S. Highway A1A
Patrick AFB FL 32925-3002

**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**

**11. SUPPLEMENTARY NOTES**

**12a. DISTRIBUTION/AVAILABILITY STATEMENT**
Approved for Public Release; Distribution is Unlimited.

**12b. DISTRIBUTION CODE**
A

**13. ABSTRACT** *(Maximum 200 words)*

This report is a computer program development specification for the Surface Wave Analysis System (SWANS). Included in the Appendix are sections on the generation of synthetic seismograms, and linear inversion theory in geophysics, and frequency variable filters.

**14. SUBJECT TERMS**
seismic surface wave analysis, geophysical inverse theory, synthetic seismogram generation, frequency variable filters, phase match filters

**15. NUMBER OF PAGES**

**16. PRICE CODE**

| | | | |
|---|---|---|---|
| **17. SECURITY CLASSIFICATION OF REPORT**<br>UNCLASSIFIED | **18. SECURITY CLASSIFICATION OF THIS PAGE**<br>UNCLASSIFIED | **19. SECURITY CLASSIFICATION OF ABSTRACT**<br>UNCLASSIFIED | **20. LIMITATION OF ABSTRACT**<br>See Block 12a |

DCS-SFS-86-02

# COMPUTER PROGRAM DEVELOPMENT SPECIFICATION FOR THE SURFACE WAVE ANALYSIS SYSTEM (SWANS)

AUGUST 1986

Authors:

K.D. HUTCHENSON
R.S. CUNFER
A.F. SUTTON

ENSCO, INC.
445 PINEDA COURT
MELBOURNE, FLORIDA  32940

## TABLE OF CONTENTS

Table of Contents (continued)

## LIST OF ILLUSTRATIONS

## LIST OF TABLES

## 1.0 SCOPE

1.1 Identification. This specification establishes the requirements for performance, design, test, and qualification of a computer program identified as the Surface Wave Analysis System (SWANS), configuration Item No. SWANS. This CPCI provides the functional requirements for the SWANS and comprises Part I of a two-part specification as defined by MIL-STD-483 (USAF).

### 1.2 Functional Summary.

1.2.1 Purpose of specification. This specification is used to define the processing techniques, graphics displays, operational parameters and procedures, and data base requirements of the SWANS. Its primary purpose is to establish the performance criteria and test criteria for which the SWANS shall be designed/developed.

1.2.2 Specification organization. This specification follows the format and content requirements of Appendix VI of MIL-STD-483 (USAF) for a Computer Program Development Specification as supplemented by MIL-STD-490 and DI-MCCR-80025 (formerly DI-E-30113/T).

1.2.3 Program functions. The SWANS program will include program functions designed to measure a surface wave station magnitude, determine path corrections between source and receiver locations, process surface waves to establish structure and attenuation along the path and to determine a scalar moment for an event across a particular path. An Adage 4250 Interactive Graphics System (IAGS) will be utilized to provide an operator interface to the SWANS for control of these tasks. Basic program functions are summarized in the following subparagraphs.

1.2.3.1 SWANS executive. The SWANS executive will provide the interface to the host system computer and will include various additional functions such as program initiation and termination, primary processing display generation and decoding, and basic control interfacing required of other program functions.

1.2.3.2 Data base management. The data base management segment includes the input/output (I/O) modules that will provide WBASE file processing functions such as open/close, read/write, and special update functions.

1.2.3.3 Program MENU and HELP's. The MENU/HELP segment will provide function MENU descriptions or definitions and will include special HELP displays that describe parameters and processing criteria for the more complex processes.

1.2.3.4 Signal review and pre-processing. Signal review supports event acquisition and signal editing functions together with special automated measurement, marking, and waveform alignment processes. Signal pre-processing will include operator controlled processes that perform special waveform quality control checks.

1.2.3.5 Retention of processing results. Waveform measurements and processing results are stored with the event of interest to permit later access for special studies or review without having to re-process the data.

## 2.0 APPLICABLE DOCUMENTS.

The following documents of the exact issue shown form a part of this specification to the extent specified herein. In the event of conflict between the documents referenced herein and the contents of this specification, the contents of this specification shall be considered a superseding requirement. However, this specification shall not negate requirements of system or higher level CI specifications.

Specifications:

Military:

MIL-S-83490; Specifications, Types and Forms, 30 October 1968.

Standards:

Military:

AFTAC-STD-2167; Air Force Technical Applications Center (AFTAC), Standard Comaputer Program Development, 1 October 1985 (supersedes AFTAC-STD-171, 15 Oct 84).

DoD-STD-2167; Defense System Software Development, 4 June 1985 (supersedes DoD-STD-1679A, 22 Oct 83 and MIL-STD-1644B, 2 Mar 84).

MIL-STD-461A; Electromagnetic Interference Characteristics Requirements for Equipment, 3 July 1973.

MIL-STD-483 (USAF); Configuration Management Practices for Equipment, Munitions, and Computer Programs, 21 March 1979.

MIL-STD-490; Specification Practices, 30 October 1968.

MIL-STD-1472B; Human Engineering Design Criteria for Military Systems, Equipment and Facilities, 10 May 1978.

MIL-STD-1521B; Technical Reviews and Audits for Systems, Equipments, and Computer Software, 4 June 1985 (supersedes MIL-STD-1521A, 1 Jun 76).

MIL-STD-1753; FORTRAN, A Supplement to American National Standards Institute X3.9-1978, 9 November 1978.

Other Government Activity:

DoD 5220.22-M; Industrial Security Manual for Safeguarding Classified Information, 17 April 1980.

Other publications:

Department of the Air Force, 1035 TCHOG (AFSC);

Requests for the following publications should be addressed to the Air Force Technical Applications Center, 1035 TCHOG/TG, Patrick Air Force Base, Florida 32925.

DCS-SFS-81-55; Computer Program Development Specification for the Seismic Event Identification System (SEIS), 12 February 1982.

DCS-SFS-82-13 Computer Program Product Specification for the Seismic Event Identification System (SEIS), 25 March 1982(revised 15 March 85).

American National Standards Publications;

Requests for the following publications should be addressed to American National Standards Institute, Inc., 1430 Broadway, New York, New York 10018.

ANSI X3.4-1977 Code for Information Interchange, 9 June 1977.

ANSI X3.9-1978 Programming Language FORTRAN, 3 April 1978.

IBM Reference Publications;

Requests for the following publications should be addressed to IBM Corporation, Publications Support Services, Department 812, 1133 Westchester Avenue, White Plains, N.Y. 10604.

GC20-1877 A Guide to the IBM 4341 processor, April 1979.

GC28-6596 FORTRAN IV Library Subprograms, February 1975.

H20-0205-3 System 360 Scientific Subroutine Package Version III Programmer's Manual, February, 1969.

Miscellaneous References;

(1) Adage, Inc., Adage 4250 Product Description, August 1978.

(2) Stoffa, P.L., P. Buhl and G.M. Bryan, 1974, The application of homomorphic deconvolution to shallow-water marine seismology - Part I: Models, Geophysics, 39, pp. 401-416.

(3) Yacoub, N.K., 1983, Instantaneous amplitudes: a new method to measure seismic magnitude, Bull. Seism. Soc. Am., 23, pp. 1345-1355.

(4) Stevens, J.L., W.L. Rodi, J. Wang, B. Srkoller, E.J. Holda, B.F. Mason and J.B. Minster, Surface wave analysis package and Shagan River to SRO station path corrections, S-Cubed, VSC-TR-82-21, April 1982.

(5) Bache, T.C., W.L. Rodi and D.G. Harkrider, 1978, Crustal structures inferred from Rayleigh-wave signatures of NTS explosions, Bull. Seism. Soc. Am., 68, pp. 1399-1413.

(6) Russell, D.R. and H.J. Hwang, 1986, Comparison of matched filters to time variable filters for amplitude determination of normal modes in surface wave analysis, Presented at 8th Annual DARPA/AFGL Seismic Research Symposium, pp. 175-182.

## 3.0 REQUIREMENTS

The developer shall design the SWANS in accordance with all requirements defined in subparagraphs herein.

3.1 Computer program definition. The Surface Wave Analysis System (SWANS) shall be a modular software system designed to measure the surface wave network magnitude, determine a path correction between selected test sites and high quality digital seismic stations, to provide the capability to process surface waves to aid in yield estimation and to determine the scalar moment for an event across a particular path. The SWANS shall be designed with the Project Office's IBM 4341 computer system and ADAGE 4250 interactive graphics system.

3.1.1 System capacities.

3.1.1.1 Program execution area. A host system core area of sufficient capacity to execute the SWANS program will be provided. It is anticipated that, in an overlay environment, a minimum of approximately one Mbytes will be required to efficiently execute program segments.

3.1.1.2 Refresh buffer. A refresh buffer within the graphics display device will be provided to support the processing displays required by the SWANS. The size of this buffer will be a minimum of 32K bytes.

3.1.1.3 Load module storage. The SWANS program load module shall be retained on a direct access storage device to allow immediate access for execution. Storage capacity requirements for this device will be dependent upon program load module size.

3.1.1.4 Data base storage. Direct access storage will be provided for the various on-line data sets required by the SWANS. These data base requirements are defined in 3.5.

3.1.1.5 System timing. System response time will be dependent upon the number of time-shared users using the host computer system and the SWANS function being performed. Programming and input/output techniques shall be incorporated which reduce, to the maximum extent possible, the response time of the SWANS to operator requests. Interrupts to the host computer system shall be minimized, to the maximum extent possible, by incorporating display self-modification functions and interrupt handling subroutines within the refresh buffer graphics order programs generated by the SWANS.

3.1.1.6 Program source code. A backup copy of all SWANS source code shall be maintained on magnetic tape or an equivalent device. Storage capacity required for this device will be dependent upon SWANS program size.

3.1.2 Interface requirements. The developer shall design the SWANS to be compatible with all interfaces defined in 3.1.2 and its subparagraphs.

3.1.2.1 Interface block diagram. The minimum essential functional interfaces of the SWANS are presented in block diagram form in Figure 3.1.2.1. The blocks and interconnections in this diagram are intended to show functional relationships between the various devices. They do not necessarily reflect actual hardware interface design.

3.1.2.1.1 Interface security. The SWANS must operate in a secure environment. All new interfaces shall be designed in accordance with the emission and susceptiblity requirements of MIL-STD-461A. Existing interfaces may be considered secure.

3.1.2.1.2 Interface summary. The SWANS will consist of a software computer program which will be executed by a host computer central processing unit (CPU) and its associated system supervisor or control program. This CPU will interface with an interactive graphics system which, through the SWANS operator and graphics devices, will provide operational control of the SWANS. Waveform and alphanumeric data files that will be used by the SWANS will reside on peripheral storage devices within the host computer system or graphics system. Primary SWANS operational messages will consist of data exchanged between the SWANS operator and the graphics system , and the graphics system and the host computer system. These data include: (a) interrupt messages, (b) graphics processing displays, (c) printer/plotter reports, (d) operator entered requests and parameters, and (e) processing results. Displays will be passed between the host system and the graphics system in the form of a graphics buffer program containing graphics orders and display data. This buffer program will be stored in a refresh buffer and used by the graphics system to refresh CRT displays or generate printer/plotter hardcopy. Buffer programs will be built within the host system by the SWANS program, transferred to the graphics system, modified by SWANS operator entries as applicable, and returned to the host system to be decoded by the SWANS program.

3.1.2.2 Detailed interface definition. The ADAGE 4250 graphics system defined in Ref. (1) shall be utilized for the graphics system. An IBM 4341 computer system equivalent to that defined in GC20-1877 shall be utilized for the Host computer system. The following subparagraphs describe the basic SWANS interfaces. More detailed interface specifications are provided in Ref. (1) and GC20-1877.

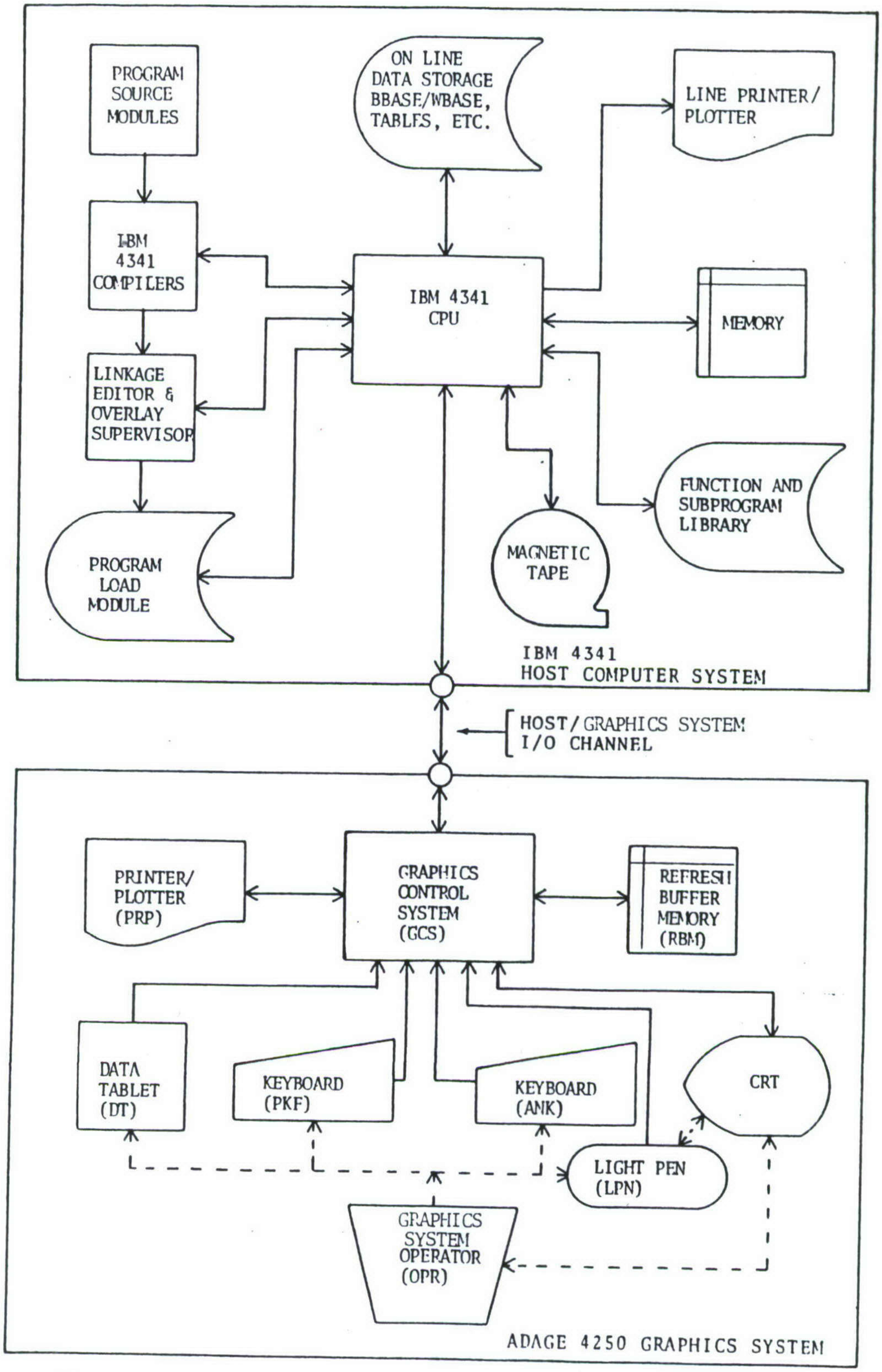
PROGRAM SOURCE MODULES
ON LINE DATA STORAGE BBASE/WBASE, TABLES, ETC.
LINE PRINTER/ PLOTTER
IBM 4341 COMPILERS
IBM 4341 CPU
MEMORY
LINKAGE EDITOR & OVERLAY SUPERVISOR
FUNCTION AND SUBPROGRAM LIBRARY
PROGRAM LOAD MODULE
MAGNETIC TAPE
IBM 4341 HOST COMPUTER SYSTEM
HOST/GRAPHICS SYSTEM I/O CHANNEL
PRINTER/ PLOTTER (PRP)
GRAPHICS CONTROL SYSTEM (GCS)
REFRESH BUFFER MEMORY (RBM)
DATA TABLET (DT)
KEYBOARD (PKF)
KEYBOARD (ANK)
CRT
LIGHT PEN (LPN)
GRAPHICS SYSTEM OPERATOR (OPR)
ADAGE 4250 GRAPHICS SYSTEM

FIGURE 3.1.2.1 System Interface Block Diagram.

3.1.2.2.1 Interactive graphics system.

3.1.2.2.1.1 SWANS operator. The SWANS operator will control all SWANS functions from a graphics system display station. Interactive devices available to the SWANS operator at each station shall include a standard alphanumeric keyboard, lighted special function keys, cathode ray tube, light pen with tip switch, and a graphics data tablet with finger switch tracking device. The operator will use these devices to enter requests and parameters into the SWANS and to evaluate or modify processing results. The minimum content of the displays that the SWANS operator will use to control processing are defined in 3.2 and subparagraphs therein. All SWANS functions requiring operator response or use of the graphics system interactive devices shall be designed in accordance with the human performance/human engineering requirements defined in 3.4.

3.1.2.2.1.2 Cathode ray tube (CRT). The CRT is a device which allows complex, high-density images to be displayed. This device currently complies with the following specifications:

(a) A precision display area of at least 10" x 10" is provided.

(b) Line width does not exceed 0.020 inches.

(c) Vector linearity is within ± 1% of full scale in the precision area.

(d) A medium persistence phosphor which minimizes "smearing" is used (P40 or equivalent).

(e) Spot motion and jitter is less than 0.05% of full screen.

(f) Programmable intensity levels are provided.

(g) A 1024 x 1024 addressable matrix is provided.

3.1.2.2.1.3 Light pen (LPN). The light pen (LPN) is a hand-held, light-sensitive, high-speed photopen. The LPN contains a tip switch which can be tested programmatically. When a vector or symbol is in the LPN's field of view and the LPN is enabled, that entire vector or symbol is brightened to allow accurate operator positioning of the LPN. The LPN tip switch is activated by pushing the pen against the CRT faceplate.

3.1.2.2.1.4 Data tablet (DT). The data tablet (DT) consists of a high accuracy, two-dimensional sketching device which provides digital x-axis and y-axis input coordinates to the graphics processor. This device includes a hand-held tracking device with finger switch. This device complies with the following specifications:

(a) DT size is 11" x 17"

(b) The DT can be used in a light pen emulation mode.

(c) DT resolution is approximately 200 lines per inch.

3.1.2.2.1.5 Alphanumeric keyboard (ANK). The alphanumeric keyboard (ANK) consists of an electronic keyboard which is conveniently positioned on the display station work surface. The ANK contains alphabetic, numeric, and special character keys. A code repeat key is included. This key generates any of the possible character codes continuously at a rate of at least 15 characters per second. The ANK will be used by the SWANS operator to enter commands and parameters on the SWANS display.

3.1.2.2.1.6 Programmable function keys (PFK). The programmable function keys (PFK) interface allows the SWANS operator to execute pre-determined functions by depressing a special key programmed for that function. The PFK consists of lighted switches mounted in an enclosure that can be conveniently positioned on the display station work surface.

3.1.2.2.1.7 Refresh buffer memory (RBM). The refresh buffer memory (RBM) is used to store display image data. The design of the SWANS shall permit refreshed displays to be presented to the SWANS operator while new display buffer programs are being built by the SWANS program.

3.1.2.2.1.8 Printer/plotter (PRP). The printer/plotter (PRP) device provides hardcopy output of CRT displays, processing results, and special reports. The PRP is a dry process electrostatic device co-located with the graphics system display station and complies with the following specifications:

(a) Plotting resolution is approximately 200 points per inch.

(b) The PRP is capable of plotting at a minimum of 1.0 inch per second for high resolution graphics and 300 lines per minute for characters.

The SWANS developer shall incorporate a backup plotting capability utilizing the Host system plotter to cover outage periods of the local plotter.

3.1.2.2.1.9 Graphics control system (GCS). The graphics control system (GCS) provides the interface and control functions for the overall graphics system. Functions performed by the GCS include, but are not limited to, the following:

(a) Character generation.

(b) Vector plotting.

(c) Display refreshing.

(d) Interrupt detect and handling.

(e) Symbol tracking.

(f) Incremental vector display drawing (plotting) for display waveforms.

3.1.2.2.2 Host computer system.

3.1.2.2.2.1 Central processing unit (CPU). The central processing unit (CPU) is the device which fetches and executes the SWANS program instructions. The CPU complies with the following specifications:

(a) Standard arithmetic operations are provided including multiply/divide.

(b) Floating point data processing is provided with an accuracy of at least six decimal places for single precision and at least twelve decimal places for double precision processing.

(c) The CPU is capable of processing 16-bit and 32-bit integer data both internally and as input/output.

3.1.2.2.2.2 Host computer operating system (HOS). The host computer operating system (HOS) consists of a system supervisor or control program that together with the host system CPU controls execution of the SWANS program. The HOS supports overlay structured programs and incorporates the required linkage editors or loaders.

3.1.2.2.2.3 Function and subprogram library (FSL). The function and subprogram library (FSL) consists of mathematical subprograms and service subprograms supplied with the host computer system to support the programming language(s) used. These subprograms consist of often used functions such as square root, log functions, sine, cosine, etc. Library subprograms are defined in GC28-6596.

3.1.2.2.2.4 Random access memory (RAM). Random access memory (RAM) provides the computer core area required for execution of the SWANS program and storage areas for internal tables and parameters used by the SWANS. Word sizes, capacity, and access times for this memory are considered compatible with the SWANS processing requirements defined in 3.2 and subparagraphs therein.

3.1.2.2.2.5 Software compiler(s). Software compiler modules (SCMs) consist of host system programs used to convert the SWANS program source code statements into the object code format required by the host computer for execution. Compilers capable of compiling the programming languages defined in 3.3.1.9 and subparagraphs therein are incorporated within the host system.

3.1.2.2.2.6 Program load module (PLM). The program load module (PLM) consists of the executable SWANS program as built by the host system linkage editor or loader.

3.1.2.2.2.7 On-line data storage (OLS). The OLS consists of on-line, direct access device storage used for the various data sets required by the SWANS.

3.1.2.2.2.8 Data base archives (DBA). The data base archives (DBA) contain the seismic event, phase, and waveform archives data required by the SWANS. These data bases include both real-time and historical archives. Current storage devices include both direct access disk and magnetic tape. Data base requirements are defined in Section 3.5 and subparagraphs therein.

3.2 Detailed functional requirements. The SWANS program shall be designed to incorporate the processing functions and techniques herein defined. The program shall be designed to permit operation in an interactive mode from an Adage 4250 interactive graphics system display console. Figure 3.2 depicts the overall functional flow anticipated for routine event processing.

Processing functions in Figure 3.2 are grouped into three basic sections. The first section includes "waveform review/edit" and "instrument response correction." The waveform review/edit is identical to that in the SEIS. This function allows both interactive and automated phase marking and alignment of seismic phases. The next function provides instrument response removal from the waveforms.

The second section includes techniques to isolate the mode of interest, both amplitude and phase, from the waveform. Interaction by the operator is allowed to help the automated techniques. As a result of the "multiple filter" and "phase-matched filter" techniques, the corrected group and phase velocity dispersion curves are available to determine the structure of a particular path by inversion. In addition, a station spectral magnitude can be calculated using the isolated phase.

The third section provides the capability to invert for earth structure, calculate the attenuation (Q structure) and the scalar moment from a simultaneous inversion for a particular path, and generate the synthetic seismograms. Once the synthetics are generated, they are then compared with the observed seismograms to determine how well the path has been modeled.

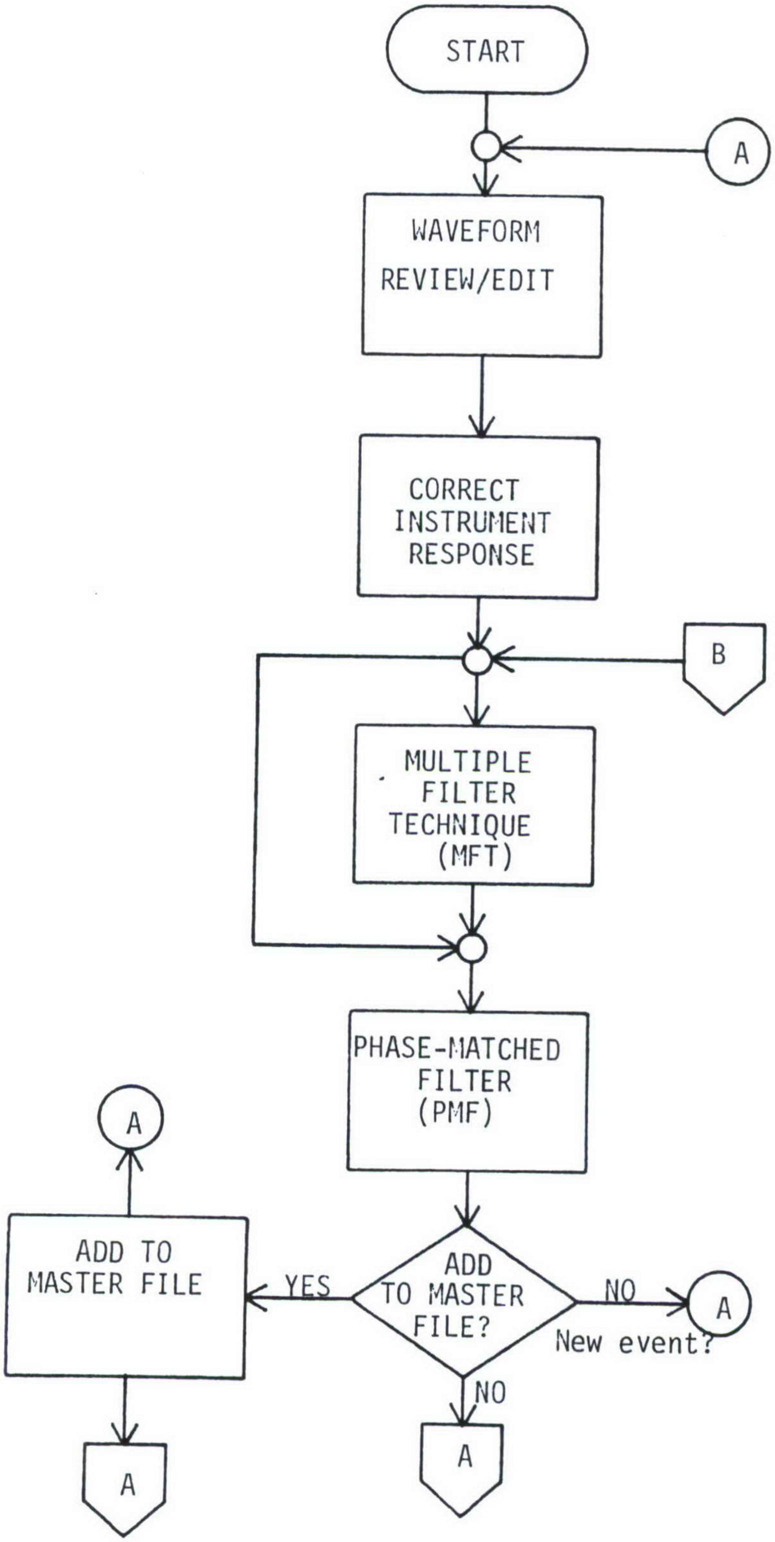

Figure 3.2 Flow Diagram of the Surface Wave Analysis System (SWANS) (Part 1 of 2)

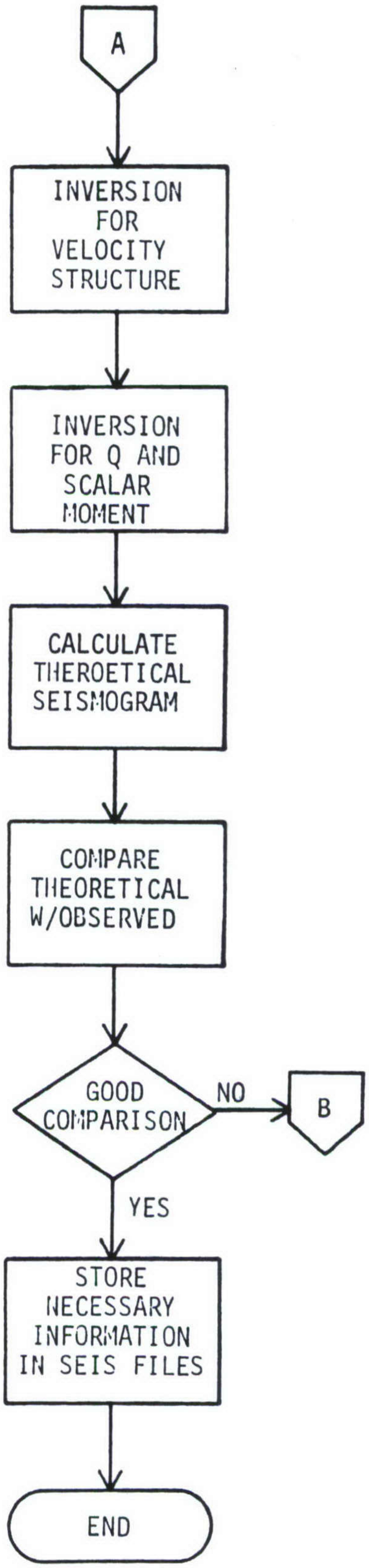
A
INVERSION FOR VELOCITY STRUCTURE
INVERSION FOR Q AND SCALAR MOMENT
CALCULATE THEROETICAL SEISMOGRAM
COMPARE THEORETICAL W/OBSERVED
GOOD COMPARISON
NO
B
YES
STORE NECESSARY INFORMATION IN SEIS FILES
END

Figure 3.2 Flow Diagram of the Surface Wave Analysis System (SWANS) (Part 2 of 2)

3.2.1 SWANS executive. The SWANS executive provides the primary interfaces between the SWANS program and the IBM 4341 host computer system and between the SWANS and the Adage 4250 graphics system. The executive also provides the control functions required to coordinate access to, and execution of, the separate major processing tasks or program segments. The executive includes functions such as program initiation, primary control display (PCD) generation, parameter initialization, menu/help display generation, and program termination.

3.2.1.1 Inputs. Inputs to the SWANS executive include control information provided by the SWANS operator via interactive displays and the special control functions or information generated internally within the SWANS program. Operator inputs to the SWANS program initiation function shall consist of the graphics system/Host System Remote Job Entry (RJE) entries and procedures in effect at the time of development.

3.2.1.2 Processing.

3.2.1.2.1 Program initiation. Standard program initiation functions such as variable and parameter initialization, opening of data sets, etc., are considered as routine programming considerations to be determined by and incorporated by the SWANS developer. The developer shall ensure that all necessary program initiation functions are accomplished by the SWANS. After appropriate initiation has been completed, the SWANS shall build the initial SWANS processing display and transfer control to the SWANS operator via the graphics system.

3.2.1.2.2 Primary Control Display (PCD). The PCD is herein defined as that portion of the SWANS interactive display which contains program control information and identification fields common to the overall program. This portion of the display contains information such as program title, input event identification, and control fields for frequently used functions such as paging, parameter access, and help display access. The basic graphics orders for the PCD display shall be generated during program initiation and, if feasible, retained in the refresh buffer thereafter to permit immediate access during subsequent processing. The basic control portions of the display shall incorporate the minimum control fields and display structures defined in the following subparagraphs. The display shall be designed for maximum readability and operator interactive efficiency.

3.2.1.2.2.1 Display title. The display title "SURFACE WAVE ANALYSIS SYSTEM (SWANS)" shall be displayed at the top of the display image.

3.2.1.2.2.2 Function selection. A function control field shall be provided which permits the SWANS operator to define functions via the alphanumeric keyboard. Except as otherwise specified herein, operator entries shall be retained thereafter. The programmable functions (PFK), also frequently referred to as special functions keys (SFK), shall be incorporated, when feasible, to maximize efficiency and to provide high-speed access to program functions.

3.2.1.2.2.3 Common function control. Where feasible, and to the maximum extent possible, all SWANS functions shall be designed to be executable from the PCD by enabling display of additional control and parameter fields as required for each of the separate functions. To the maximum extent possible, such display subsets shall be built once for often used functions and retained thereafter either in core, or within the refresh buffer in a blanked mode, to permit immediate access for display.

3.2.1.2.2.4 Data identification control fields. The PCD shall contain control fields that permit the SWANS operator to define the data base divisions and files desired. The operator shall also be permitted to specify a specific station or channel.

3.2.1.2.3 Reset feature. The SWANS shall incorporate a reset function that resets all SWANS control and parameter fields to the defaults assumed at program initiation. This feature will permit the SWANS operator to rapidly recall default values before starting processing of a new event when special controls or parameters were entered for the previous event.

3.2.1.2.4 Executive control. The SWANS executive shall maintain primary control over all linkage functions required to access any major SWANS program segment (i.e., no segment subordinate to the executive segment shall issue a LINK, ATTACH, or similar system instruction).

3.2.1.2.5 SWANS menu display. The SWANS shall, at operator request, build a menu display containing a listing of all SWANS functions including an appropriate display title. A display paging function shall be provided if the size of the menu warrants. The menu listing shall include a brief description of each of the listed functions.

3.2.1.2.6 SWANS help displays. The SWANS shall, at operator request, generate special help displays for all program functions that require operator definition of input parameters. When feasible, functions should be grouped (within the parameter and help displays) to minimize the number of separate help displays required. Help displays shall, as a minimum, include the following:

(a) Brief abstract of the program function(s),

(b) Special options (if any), and

(c) Parameter ranges and defaults.

3.2.1.2.7 Help accessibility. All Help displays shall be made available for display as part of the Menu function defined in 3.2.1.2.5 and the appropriate individual help(s) shall also be made available during the actual processing to which they apply.

3.2.1.2.8 Program termination. The design of the SWANS shall permit the SWANS operator to terminate SWANS processing from any SWANS processing display by a display entry of the commands END or STOP via the alphanumeric keyboard. Termination shall not be permitted by use of an individual or special function key. This limitation shall be incorporated to prevent accidental program termination. Upon receipt of a STOP or END command, the SWANS shall perform the following subfunctions:

(a) Store applicable internal tables or parameters on the appropriate storage devices,

(b) Close all applicable SWANS data sets, and

(c) Return processing control to the host operating system.

3.2.1.2.9 Graphics display and buffer wipeout. As a part of program termination, the SWANS shall execute a display and refresh buffer wipeout function equivalent to that employed in the Project Office graphics monitor program TGSBROM.

3.2.1.3 Outputs. Outputs from the SWANS executive include control information passed to the host system or graphics system, graphics order buffer programs or graphics program subsets passed to other program segments or the display device, and error messages (when appropriate) passed to the operator via appropriate displays.

3.2.2 Data base management. The data base management (DBM) function incorporates the subfunctions required to access the Waveform Base (WBASE), B-System Station Locations (BSTALOCS), and other on-line data bases required for execution.

3.2.2.1 Inputs. Inputs to the DBM function include, but are not limited to, items such as the following:

(a) processing commands provided by internal program functions or by the operator,

(b) event and phase information provided by other program functions or acquired from the data base being accessed, and

(c) the data base to be accessed.

3.2.2.2 Processing.

3.2.2.2.1 WBASE Input/Output. This subfunction shall perform the required open, close, read, and write operations for the WBASE data base. To the maximum extent possible, the SWANS developer shall adapt software modules from the Project Office programs BDAMP, BROM, WPS, HIEDS or SEIS to perform these subfunctions.

3.2.2.2.2 BSTALOCS input. The BSTALOCS access function shall provide access to the BSTALOCS file for input operations only. To the maximum extent possible, the developer shall adapt existing software from the Project Office programs TGSHYPO7, and SEIS (see TGSHYPO7 and DCS-SFS-81-55), as applicable, to perform this function.

3.2.2.2.3 Storage of measurements. The SWANS shall provide a method for storing measurement results with the subject event and shall provide a method for subsequent access to the measurements.

3.2.2.2.4 Archives access. The SWANS developer shall incorporate existing methods for archiving processed events and for reloading applicable disk data bases from the archives. It is not mandatory that the archiving or reload functions be interactive. Existing Project Office batch programs will be utilized or adapted (ARCHIV and MAINPT). Archives shall be maintained in WPS "load tape" format (see TR-77-2(c)).

3.2.2.3 Outputs. Outputs from the data base management function will include waveform data, event records, phase records, station information, and updated or modified data bases.

3.2.3 Signal review and pre-processing. The SWANS signal review and pre-processing function is identical to the SEIS REVIEW function. This function supports processes such as a signal review and editing, automatic phase marking and alignment. Provisions are included for interactive timing and measurement of event phases and operator identification of clipped data.

The SWANS processing functions will only operate on long period data. The full capability to handle both short and long period data has not been removed from the REVIEW function.

3.2.3.1 Inputs. Inputs to the signal review and pre-processing function include, but are not limited to the following:

(a) event, phase and waveform data,

(b) applicable input file and data identification information,

(c) applicable phase arrival time information, and

(d) applicable data editing information.

3.2.3.2 Processing.

3.2.3.2.1 Binary gain ranging conversions. Binary gain ranging (BGR) is a technique which permits a large dynamic range to be preserved on a single seismic waveform data channel while retaining a 16-bit integer word format which reduces data storage requirements. Waveform data that are recorded in BGR format must be converted to normal integer or floating point format prior to subsequent processing functions. The BGR conversion function shall perform this conversion process automatically whenever BGR data is detected during waveform data input processing.

3.2.3.2.1.1 BGR parameters. Current TG load tape archives contain a data type control word in word two of the waveform headers. This control word contains the integer value 1 if the data are in BGR format.

3.2.3.2.1.2 BGR format. A BGR data word consists of a 16-bit integer word with an exponent value in bits 0-3 and a scaled data value (including sign bit) in bits 4-15.

Table 3.2.3.1 Binary Gain Ranging (BGR) Word Format.

16-bit word

| | Exponent | | Data Value | |
|---|---|---|---|---|
| bits | 0 | 3 | 4 | 15 |

Data value ($D_1$) has been scaled by:

$$D_1 = \frac{V \cdot SF}{2^{(S-G)}}$$

where; $D_1$ = Scaled data value,

$G$ = exponent

$S$ = exponent scale factor

$SF$ = Normally 1.0 (waveform segment header scale factor).

$V$ = original data value.

3.2.3.2.1.3 BGR conversion equation. All data points of the BGR format waveform shall be converted by the following equation:

(a) $D_1 = V \cdot 2^{(S-G)} \cdot SF \cdot SF_2$

where; $D_1$ = converted value,

$G$ = exponent value (bits 0-3 of the BGR word),

$S$ = exponent scale factor (normally 10 or 0),

$SF$ = waveform segment header scale factor (normally 1.0 for BGR data).

$SF_2$ = 1.0 (if floating point output) or 32767.0/(maximum value)(if integer output), and

$V$ = BGR data value (bits 4-15 of the BGR word).

NOTE: $SF_2$ is used to maintain maximum amplitude resolution when 16-bit integer, it is later multiplied by $1.0/SF_2$ if measurements are required on the data. Normally, 16-bit integer output is used for "display only" functions and floating point output is used for measurements.

3.2.3.2.2 Directory Scan. At operator request, the SWANS shall provide a display list of the specified file directory contents including, as a minimum, the station, channel identifier, start time, and end time of each available data channel.

3.2.3.2.3 Waveform Scan and Review. The waveform scan and review function shall provide the operator with a method for visually inspecting the raw input waveform data. The SWANS shall generate waveform displays of all applicable channels to permit review by the SWANS operator.

3.2.3.2.3.1 Scan time base. The SWANS shall display short period waveform segments of approximately 40 seconds using a time base of approximately 5mm/sec and long period waveform segments of approximately 13 minutes 20 seconds using a time base of approximately 15mm/min.

3.2.3.2.3.2 Automatic marking and alignment. When short period P-phase waveform data and associated arrival times are available, the SWANS shall automatically mark and align these phases such that approximately 5 seconds of noise precedes the P-phase on the display. When long period LR phase waveform data and associated arrival times are available, the SWANS shall automatically mark and then align these phases to the approximate center (x-axis) of the display.

3.2.3.2.3.3 Waveform annotation. As a minimum, the station channel identifier, LR or P-phase arrival time, and waveform start time shall be included as annotation for each displayed waveform.

3.2.3.2.3.4 Processing window definition. A method shall be incorporated to mark the start of the processing window on long period data which shall be used by the remainder of the SWANS. This identifier shall be distinct from seismic phase identifiers.

3.2.3.2.4 Waveform select/delete. The waveform scan display shall provide a method by which the SWANS operator can flag any waveform for selection or deletion. The SWANS shall preserve the most recent waveform selection, or deletion.

3.2.3.2.5 Phase marking. The SWANS shall provide a capability for interactive operator marking and identification of at least four seismic phases per short period waveform channel and three seismic phases per long period waveform channel. Interactive editing shall permit modification of phase identifiers as well as repositioning or deletion of phase scribe marks (the P and LR phases and can not be deleted). The SWANS shall retain the most recent phase identification and scribe mark information for subsequent SWANS processing functions.

3.2.3.2.6 Scan/Edit display paging. When multiple display pages are required, random access to any desired page shall be provided.

3.2.3.2.8 WBASE subfile updates. The SWANS shall incorporate both a phase subfile update and an event subfile update. Applicable event and phase arrival information obtained from SWANS processing shall be stored in these WBASE subfiles.

3.2.3.2.9 Response time. If feasible, acquisition of waveforms and waveform measurement functions shall be overlapped with operator analysis of displays to reduce Host system response time to operator requests.

3.2.3.2.7 Identification of clipped data. A SWANS quality control function shall provide the operator with capability to interactively identify invalid or clipped data. This quality control information shall be preserved for a quality control report and for subsequent SWANS processing functions.

3.2.3.3 Outputs. The basic outputs from the signal review and pre-processing function consists of an edited event together with applicable timing information. Timing shall be retained, when appropriate, for use by other SWANS functions and for subsequent addition to a measurements file or subfile. Only long period data will be accepted as input to the SWANS processing functions.

3.2.4 System Response Correction. The SWANS system response correction function removes the instrument response, both amplitude and phase, from the selected long period waveform.

3.2.4.1 Inputs. Inputs to the system response correction function shall include, but not be limited to, the following:

(a) A digital long-period waveform channel selected by the operator while in the waveform review/edit function, and the

(b) Instrument response information.

3.4.2.4 Spectrum Computation. A basic Fast Fourier Transform (FFT) equivalent to the Cooley-Tukey method shall be incorporated to provide the spectrum of the input time series. Both forward and inverse FFT's will be computed.

3.2.4.2.2 Instrument Response Removal. The instrument amplitude and phase responses shall be removed from the raw input spectrum to obtain the corrected spectrum and time series. The application of the applicable instrument response will be automated.

3.2.4.3 Outputs. Outputs from the system response correction function shall include, but not be limited to, the waveform corrected for instrument response, to be used by other SWANS program segments.

3.2.5 Multiple Filter. The SWANS multiple filter function computes and displays the group velocity of dispersed seismic signals.

3.2.5.1 Inputs. Inputs to the multiple filter function shall include, but not be limited to, the following:

(a) A digital long-period waveform segment selected by the operator while in the waveform review/edit function, and the

(b) Long-period waveform window and Rayleigh phase time information.

3.2.5.2 Processing. The multiple filter processing is defined in the following paragraphs. When applicable, standard default parameter values shall be automatically provided by the program.

3.2.5.2.1 Mode identification and editing. The multiple filter function shall provide a method to display the various levels of energy as a function of group velocity (km/sec) and period (seconds) to aid in the identification of the fundamental group velocity mode. The fundamental mode will initially be automatically identified by the highest energy levels in group velocity/period space. A method shall be available to interactively mark the mode of interest.

3.2.5.2.2 Window definition. The operator shall have the capability to define the limits in group velocity/period space in which to identify the mode of interest.

3.2.5.3 Outputs. Outputs from the multiple filter shall include, but not be limited to, the following:

(a) Group velocity/period plots, and a

(b) Group velocity dispersion curve.

3.2.6 Match Filter. The SWANS phase-matched filter function computes and displays the phase-matched filter found from initial estimates provided by the group velocity dispersion curve. The purpose of this function is to isolate a single surface wave normal mode from path effects such as multipathing and higher mode contamination.

3.2.6.1 Inputs. Inputs to the phase-matched filter function shall include, but not be limited to, the following:

(a) A digital long-period waveform segment,

(b) Initial estimates of the group velocity, and

(c) Initial estimates of the number of breakpoints desired when applying the cubic spline interpolation.

3.2.6.2 Processing. The phase matched filter processing is defined in the following paragraphs. When applicable, standard default parameter values shall be automatically provided by the program.

3.2.6.2.1 Spectrum computation. A basic Fast Fourier Transform (FFT) equivalent to the Cooley-Tukey method shall be incorporated to provide the spectrum of the time series. Both the forward and inverse FFT's will be computed to provide amplitude and phase spectra.

3.2.6.2.2 Interpolation. A least-squares cubic spline interpolator will be used to smooth initial and later estimates of both group and phase velocity. Interaction will be available to control the amount of smoothing.

3.2.6.2.3 Phase unwrapping. An algorithm based on Stoffa et al. (1974) will be used to provide smooth estimates of the phase spectrum. This is accomplished by numerically differentiating $\Delta\phi$ with respect to frequency ($\omega$), then numerically integrating over $\omega$. This technique avoids $2\pi$ discontinuities by not explicitly using the arctangent function.

3.2.6.2.4 Time variable filtering. An algorithm based on Russell and Hwang (1986) is included as an option to reduce spectral amplitude distortion, which is caused by time windowing the phase-matched filtered waveform.

3.2.6.3 Outputs. Outputs from the phase-matched filter function shall include, but not be limited to, the following:

(a) Phase velocity dispersion curve (or phase-matched filter),

(b) Corrected group velocity dispersion curve, and the

(c) Corrected spectrum of the isolated mode of interest.

3.2.7 Inversion. The SWANS shall determine both the theoretical velocity and Q structure for a given source-receiver path. The corrected phase and group velocities, determined previously from multiple and phase matched filter, and the spectrum of an event(s) traveling across the path are required to calculate the earth structure. The synthetic seismogram and earth structure are required to calculate the attenuation (or Q) structure.

3.2.7.1 Inputs. Inputs to the inversion process shall include as required, but not be limited to, the following:

(a) Corrected group and phase velocities for the path,

(b) The spectrum of an event between the particular source-receiver path (or a statistical average of several events with the standard deviation),

(c) Source structure for refined theoretical calculation of amplitude spectra (Bache *et al*., 1978), and

(d) For the Q inversion, the inverted velocity structure and the observed amplitude spectrum for the paths shall be required.

3.2.7.2 Processing. The inversion processing is defined in the following paragraphs. When applicable, standard default parameter values shall be automatically provided by the program.

3.2.7.2.1 Damping. Options shall be provided to calculate an inversion for earth structure using both stochastic and differential damping in the inversion process.

3.2.7.2.2 Inversion types. Two functions shall be provided; an inversion for velocity structure and an inversion for Q structure.

3.2.7.2.2.1 Inversion for velocity structure. A function shall be provided to use the corrected group and phase velocities and the signal characteristics for a particular source-receiver pair to calculate the average velocity structure across the path.

3.2.7.2.2.2 Inversion for attenuation/scalar moment. A function shall be provided to use the inverted velocity structure obtained from previous processing and the observed amplitude spectrum to calculate the scalar moment and Q structure simultaneously (see paragraph 3.2.11).

3.2.7.3 Outputs. Outputs from the inversion process shall include, but not be limited to, the following:

(a) Theoretical earth structure in terms of velocity, Q and density as a function of depth,

(b) Resolving kernels which show the resolution at the various depths examined.

(c) Standard deviation of the earth structure, and

(d) Scalar moment.

3.2.8 Synthetic seismograms. The SWANS shall incorporate a method to generate synthetic seismograms based on the corrected group and phase velocities and the theoretical earth model.

3.2.8.1 Inputs. Inputs to the synthetic seismogram process shall include, but not be limited to, the following:

(a) Theoretical earth model velocity and Q structure and density as a function of depth for the path.

3.2.8.2 Processing. Processing for this function shall include those functions necessary to construct a synthetic signal from the input constraints on the signal, defined in 3.2.8.1.

3.2.8.3 Outputs. Outputs from the synthetic seismogram function shall include, but not be limited to, the time-series of the synthetic waveform which can then be directly compared with the actual waveform observed for a particular path to determine how well a particular path has been understood.

3.2.9 Master filters. The SWANS shall incorporate a method to generate average filters (i.e. master filters) for a particular source-receiver pair. The master filters shall be constructed from previous SWANS processing results.

3.2.9.1 Inputs. Inputs to the master filter process shall include, but not be limited to, the following:

(a) Corrected group and phase velocities, and the

(b) Required spectrum information.

3.2.9.2 Processing. The processing shall include an averaging process to combine the results from several events crossing a particular source-receiver pair into an average result. The ability shall exist to add additional events to the average.

3.2.9.2.1 Average spectrum. The capability shall exist to perform an average of the input events for a particular path. The process shall be calculated as follows:

$$u(\omega) = 1/N \sum_{i=1}^{N} u_i(\omega)$$

where;

$u_i(\omega)$ = individual spectra,

N = number of events.

3.2.9.2.2 Average group and phase velocity. The capability shall exist to perform an average of the input group and phase velocities for a particular path.

3.2.9.2.3 Path correction. The ensemble average (average of the spectra, group and phase velocities for events across a path) shall be inverted to obtain the theoretical seismogram for a particular path. The theoretical seismogram shall provide path corrections for a particular source-receiver pair.

3.2.9.3 Outputs. Outputs from the master filter processing shall include, but not be limited to, master filters which provide path correction for given source-receiver pairs.

3.2.10 Spectral magnitude. The SWANS shall incorporate a method to calculate a spectral magnitude for each processed station.

3.2.10.1 Inputs. Inputs to the spectral magnitude process shall include, but not be limited to, the appropriate information for the signal after isolating the phase of interest, usually the fundamental mode Rayleigh phase.

3.2.10.2 Processing. Processing for the spectral magnitude function shall include a method similar to the one defined by Yacoub (1983):

$$Ms = 1/7 \sum_{i=17}^{23} \log[\max_{LR,i}(F^{-1} A_i(\omega))]$$

where; $A_i$ = corrected amplitude spectrum,

$F^{-1}$= inverse Fast Fourier transform operator,

$\max_{LR,i}$ = local maximum instantaneous operator at period i for the appropriate group velocity of the fundamental Rayleigh phase.

3.2.10.3 Outputs. Outputs from the spectral magnitude process shall include, but not be limited to, the magnitude of the signal at a particular station.

3.2.11 Scalar Moment. The SWANS shall determine scalar moment for events of interest.

3.2.11.1 Inputs. Inputs in the calculation of scalar moment shall include, but not be limited to, the following:

(a) Equivalent inputs as required in the inversion for Q structure:

1) Average spectrum for a particular path and its standard deviation, and

2) Velocity structure for the path found from velocity inversion.

(b) Individual event spectra used in the calculation of the absolute moment.

3.2.11.2 Processing. Processing for the scalar moment involves both the calculation of a relative moment and an absolute moment. A relative moment is obtained during the initial stages in the creation of the master filter, during the averaging of events across a path. The absolute moment is obtained after the creation of the master filter for a path when the amplitude spectrum of input events are compared against the amplitude spectrum of the theoretical seismogram for the path. When applicable, standard default parameter values shall be automatically provided by the program.

3.2.11.2.1 Relative moments. The relative moment of the individual spectra are obtained by minimizing the residual ($\varepsilon$) using least-squares:

$$\varepsilon = \overline{u}(\omega) - M_N^R \, u_N(\omega)$$

where:

$$\overline{u}(\omega) = \sum_{i=1}^{N-1} M_i^R \, u_i(\omega)$$

N = number of events.

3.2.11.2.2 Standard deviation. Standard deviations of the average spectrum, $u(\omega)$, is defined as:

$$\sigma^2(\overline{u}(\omega)) = [1/(N-1)] \sum_{i=1}^{N} [\overline{u}(\omega) - M_i^2 u_i(\omega)]^2$$

where N = number of events.

3.2.11.2.3 Absolute moment. The average spectrum for a path is inverted for Q structure and attenuation-corrected moment. The absolute scalar moment is then obtained for any input event by minimizing the residual $(\omega)$ between the amplitude spectrum of the input event with the path corrected amplitude spectrum:

$$\varepsilon = m_{oi} S(\omega) - S_i(\omega)$$

where:

$S(\omega)$ = path-corrected amplitude spectra

$S_i(\omega)$ = amplitude spectrum from an input event.

3.2.11.3 Outputs. Outputs shall include, but not be limited to, the path-corrected spectrum and the absolute scalar moment for each event of interest.

3.3 Special requirements.

3.3.1 Programming methods. The computer program shall be designed to include program and data structures which enhance readability, controllability, testability, extendability, and reliability. Software previously coded under other contracts shall not be recoded for the sole purpose of conforming to the standards herein specified. However, such software shall be brought into compliance with the commenting standards herein defined. As a minimum, program design shall incorporate the following.

3.3.1.1 Definitions.

3.3.1.1.1 Software. Computer programs and computer data bases shall be considered as software.

3.3.1.1.2. Computer program structure. The computer program structure shall consist of a Computer Program Configuration Item, Computer Program Components and Modules.

3.3.1.1.2.1 Computer Program Configuration Item (CPCI). A CPCI is the actual computer program end item in the form of computer instructions stored on machine-readable media. A CPCI shall consist of one or more computer program components.

3.3.1.1.2.2 Computer Program Component (CPC). A CPC is a functionally, logically distinct part of a CPCI. A CPC is identified for purposes of convenience in specifying and developing a CPCI as an assembly of subordinate elements. A CPC consists of a logical composition of one or more subordinate or interfacing modules. A CPC may be referred to as a program segment.

3.3.1.1.2.3 Module. A module performs a complete logical process by execution of a set of instructions which have clearly defined inputs, processing logic, and output. A module is the smallest set of executable statements able to be assembled or compiled. A module shall consist of a set of instructions in a form consistent with the appropriate language, operating system, and computer.

3.3.1.2 Hierarchical program design. Programs shall be designed in a hierarchical manner and the levels of the hierarchy shall correspond to the levels of control of the tasks performed by the program. Each level of the program shall be complete by itself. The lowest level of processing shall correspond to the module. Provisions for incorporating existing modules into the hierarchy shall be made so as to maximize the reuse of previously developed software.

3.3.1.3 Execution order programming. Program components shall be programmed in execution order; that is, components at the higher levels in the hierarchical program organization shall be programmed before components at the lower levels. Specifically, calling routines shall be employed in early checkout. Dummy calling routines are not permitted. When a routine is programmed, it shall be programmed to comply with an interface that has already been programmed. When a routine is programmed, it shall always be possible to check it out with components at higher levels in the hierarchy.

3.3.1.4 Standardized logic. Logic shall be standardized in such a manner as to employ only closed logic structures in the construction of program components. Closed logic structures are structures that have a single entry point and a single exit point exclusive of initialization. Except as otherwise specified herein, all modules shall be coded as closed logic structures.

3.3.1.5 Module size. In general, a module shall contain no more than 100 executable source language statements except where necessary to avoid undue fragmentation. This size restriction shall not apply to existing software used intact from other systems. However, the developer shall, when feasible and as time permits, consider converting extremely large existing modules into smaller, more manageable module structures.

3.3.1.6 Commenting standards. Except as specified otherwise herein, software developed under this contract shall adhere to the commenting standards defined in the following subparagraphs.

3.3.1.6.1 Banners. A banner shall be a block of comments which appear once at the beginning of each module. A banner shall visually break the project software into units of code corresponding to the CPCI decomposition (module level). Banners shall have an identical format for each module within a CPC. The banner shall enclose the following information: CPCI title, CPC title, and CPC number. The banner shall immediately precede the header.

3.3.1.6.2 Headers. Headers shall consist of a block of consecutive comments arranged to facilitate the understanding and readability of each module. Except as otherwise specified herein, this form of block commenting shall be used in lieu of individual comments being scattered throughout a module. Headers shall occur once at the beginning of each module and shall conform to the standards described herein. The observer shall be able to read the MODULE-HEADER and understand the processing activities of the module without having to read program code. The minimum required MODULE-HEADER comments are described below. Except for deviations approved by the procuring activity, these comments shall appear in the form and in the order as illustrated in the following list:

(a) MODULE-NAME - the name and short title of the module shall be listed.

(b) SECURITY CLASSIFICATION - the security classification of the module shall be displayed following the module name. If the module is classified, then the security classification shall be more prominently displayed. Comments concerning the security aspects of the module, its inputs and its outputs shall be discussed more fully elsewhere in the banner.

(c) ABSTRACT - the ABSTRACT shall be a set of consecutive comments which describe the module's purpose, use and processing activities. Elaboration on the technical aspects of the algorithms should be avoided where references to external Government documentation would suffice. The ABSTRACT should paraphrase the activities of the code in English terms. References made to external Government-owned documentation shall be listed in the REFERENCES comment section.

(d) REFERENCES - NO-1, author, title, date (YY/MM/DD)
NO-2, etc.

(e) INPUTS - variables, tables (local, system), files and other data input sources shall be identified separately as to type, unit of measure, accuracy or precision requirements, and frequency of arrival.

(f) OUTPUTS - variables, tables (local, system), files, and other data output sources shall be identified in the same manner as inputs.

(g) MODULES CALLED - names of other programs called followed by a brief abstract of purpose and pre- and post-conditions of each call.

(h) LIMITATIONS - descriptions of any constraints upon the execution of the program. For instance, conditions which alter the logical operation of the program or cause the results of the program's computations to be altered.

(i) LIMITATIONS - MODULES - any constraints on the use or execution of the module shall be listed.

(j) PROGRAMMER - the name, organization and date of the original programmer shall be listed.

(k) MODIFICATIONS - NO-1, MOD description, DATE
(YY/MM/DD) NO-2, etc.

3.3.1.6.3 Special comments. Wherever code is particularly subtle or confusing, SPECIAL-COMMENTS shall precede the statement(s) to describe the activities of the subject code. SPECIAL-COMMENTS are provided only to aid the observer in reading program code and are not intended to replace MODULE-HEADER comments.

3.3.1.7 Coding conventions. Computer programs coded for the system shall employ only the control constructs listed below. These constructs shall be built using logically equivalent language simulations. Instructions in the language used shall follow the graphic representation in Figure 3.3.1.7. These constructs are defined as follows:

(a) SEQUENCE - sequence of two or more operations.

(b) IF-THEN-ELSE - conditional branch to one of two mutually exclusive operations and continue.

(c) DO-WHILE - operation repeated while a condition is true. Test is before operation.

(d) DO-UNTIL - operation repeated until a condition becomes true. Test is after operation.

(e) CASE - select one of many possible cases.

3.3.1.7.1 Limitations. Coding shall be restricted as follows:

(a) Each line of source code shall contain no more than one statement.

(b) Names of operator command, data entries, program components, variables, procedures, and other software components shall be consistent with those used in system design.

(c) Code shall be written such that no code is modified during execution. Graphics order buffer programs are specifically exempted from this requirement.

3.3.1.8 Character set standards. Character sets shall conform to standards in FIP-1 Standard Code for Information Interchange, ANSI X3.4-1977.

3.3.1.9 Programming languages. Except as otherwise specified herein, all programs shall be restricted to one or more of the following languages:

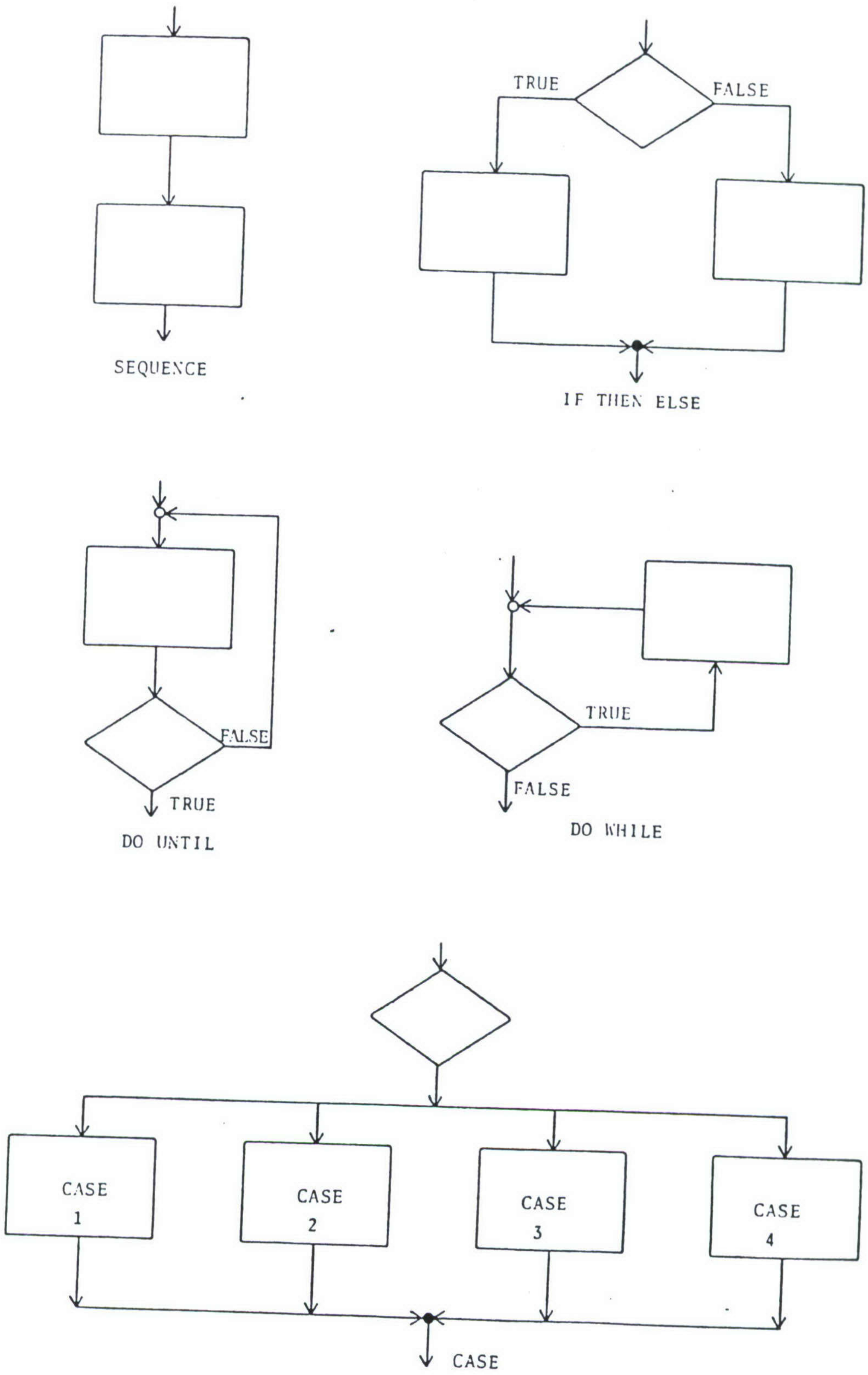

FIGURE 3.3.1.7 Coding Convention Examples.

(a) FORTRAN as per ANSI S3.9-1978 and MIL-STD-1753, and

(b) COBOL as per FIPS PUB 21-1.

The contractor shall restrict compiler features to those implementing the syntax and semantic requirements of the above specified version(s) of the approved standard(s). Any deviations from the set standard(s) shall be fully justified in terms of cost, schedule and performance benefits. Except as otherwise specified herein, written government approval shall be required before implementation of any deviations from these standards. The contractor shall minimize the use of Assembly Language source code. Except for direct access disk I/O modules, graphics order program generation modules, and graphics system or host system interface modules, the use of assembly level coding for a Computer Program Component (CPC) requires written government approval before implementation.

3.3.2 Program organization.

3.3.2.1 Program segment execution. The program segment in execution shall perform all necessary wrapup functions prior to transferring control to another program segment. Program design shall permit the operator to request any program segment from any function display in the executing segment.

3.3.2.2 Overlay structures. Overlay structures, when required, shall be designed for maximum efficiency. When feasible, all modules within a CPC shall be retained within the same overlay segment.

3.3.2.3 Classification requirements. This program will have access to classified system tables and data files. Some of the output reports will also be classified. All classified material shall be handled and stored in accordance with the provisions for protection of classified information defined in DoD 5220.22-M. Core dumps resulting form program abends shall be initially handled as SECRET material. If warranted, classification downgrading may be performed after a thorough review of the core dump contents.

3.3.3 Expandability. Program design shall incorporate provisions for future expansion.

3.3.4 Special timing. Program design shall incorporate processing and program structuring techniques which maximize processing efficiency and minimize process execution times.

3.3.5 Error recovery. The developer shall incorporate error recovery processes and techniques within new program software which will minimize:

(a) Program abends due to invalid operator entries,

(b) Invalid processing results due to other improper parameters, and

(c) Host computer abnormal termination of the program for other causes when such termination could be prevented by utilization of available programming techniques or procedures.

When appropriate, error messages shall be displayed informing the operator of the error condition.

3.4 Human performance. The overall design of the program shall incorporate human engineering design principles in accordance with MIL-STD-1472B which maximize the effectiveness of man-equipment combinations and minimize demands upon human skill, training, and manpower.

3.4.1 Operator response. Program functions shall be implemented in a manner which permits the operator to initiate requests and respond to or analyze outputs in rapid and efficient manner. Design procedures and methods which reduce operator fatigue shall be incorporated.

3.4.2 Program response. The program shall be structured in a manner which permits high-speed access to program functions and data. Except as otherwise specified herein, program response to an operator request shall not exceed approximately 15 seconds maximum CPU time. Processing of an individual station or channel of an event shall be permitted to exceed this maximum provided operator analysis of intermediate results requires more than 15 seconds and analysis can be overlapped with CPU processing of the next waveform segment, station or channel.

3.4.3 Graphics displays. All graphics displays shall be designed for maximum readability. To the maximum extent possible, light pen, graphics tablet, and special function keys shall be used to control program functions in lieu of keyboard entries. When keyboard entries are required, the display design shall permit rapid access to, and efficient entry of the required information. The number of processing control displays shall be minimized. Additional control displays shall not be added when an existing display can feasibly be utilized. Displays shall be refreshed at a refresh rate of approximately forty frames per second or higher when possible. High density images displayed simultaneously on two or more Adage display stations may result in flicker. Display flicker shall be judged on the basis of flicker visibility when a single display station is operating.

3.5 Data base requirements.

3.5.1 Sources and types of input. The SWANS will require access to the on-line direct access data sets described in the following subparagraphs.

3.5.1.1 WBASE File. The WBASE file contains event, phase, and waveform data as defined in the TR-77-1.

3.5.1.2 Station Location File. The Station Location file shall consist of the Project Office BSTALOCS station location file. This is an indexed sequential data set containing station codes, station locations, and station elevations. The station location coordinates are stored in values of geocentric co-latitude and east longitude. This file requires approximately 145K bytes of on-line direct access storage, allowing approximately 24 bytes per station record.

3.5.1.3 Instrument Responses. The Instrument Response file shall contain the amplitude and phase responses of any applicable instruments. The file will consist of a directory containing the instruments available in the file and allowing random access to the actual response curves.

3.5.1.4 Master filter file. The Master Filter file shall contain the necessary information to average an ensemble of events for the purpose of constructing well-constrained earth structures for a particular path. This data base shall be built and modified by the SWANS.

3.5.2 Destination and types of outputs. Outputs from the SWANS will consist of alphanumerics, digital waveforms, and reports which may be a combination of alphanumeric and waveform data. The output destinations will include:

(a) WBASE files,

(b) A printer, plotter, for hard-copy output of reports, and

(c) The graphics system display device.

3.5.3 Internal tables and parameters. The SWANS shall use internal tables and parameters to accomplish the objectives of the program. Internal tables will only be used when separate, on-line tables are not available.

3.6 Adaptation requirements. This paragraph is not applicable to this specification.

3.7 Government-furnished property list. Where feasible and to the maximum extent possible, the SWANS developer shall utilize or adapt (if appropriate) applicable program modules from existing software which performs the needs and requirements of the SWANS.

## 4.0 QUALITY ASSURANCE PROVISIONS

4.1 Introduction and definitions. The SWANS shall be tested and evaluated to verify its compliance with all requirements of this specification. Development Test and Evaluation (DT&E) shall be performed in accordance with test plans and procedures prepared by the developer and approved by the procuring activity. These plans and procedures shall be developed in accordance with the qualification and test requirements defined herein and the developer shall participate as a member of the test force.

4.1.1 Test types. DT&E shall incorporate the four basic types of testing described in the following subparagraphs. Defects or errors encountered at lower levels of testing shall be corrected before higher level tests are started.

4.1.1.1 Computer programming test and evaluation. Computer programming test and evaluation (CPT&E) is usually conducted prior to or in parallel with preliminary or formal qualification tests. CPT&E is oriented primarily towards support of the design and development process. CPT&E includes, but is not limited to, normal debug techniques, code walk throughs and independent verification of algorithms. This testing is normally performed by the developer in direct support of the design, code and checkout, and integration and phases of development. These test may be conducted informally, but all test procedures, input data, and test results shall be made available for review by the procuring activity.

4.1.1.2 Preliminary qualification tests. Preliminary qualification test (PQT) is oriented primarily towards verifying compliance with specification requirements for portions of the program prior to fully integrated/formal qualification. PQT shall be performed at the procuring activity facility, shall incorporate the graphics system interface. PQT shall be considered as final contractor testing just prior to FQT.

4.1.1.3 Formal qualification tests. Formal qualification test (FQT) is oriented primarily towards testing of the fully integrated program using procuring activity facilities. FQT includes, but is not limited to, verification of compliance with all physical interface requirements, and demonstration that the installation process has not degraded performance from that demonstrated in earlier tests. All FQT shall be witnessed by the government or government designated personnel.

4.1.1.4 System test. System test is not applicable to the SWANS. The SWANS is comprised of a single CPCI and final testing of the integrated CPC's and segments shall be accomplished during FQT.

4.1.2 Verification methods. DT&E shall incorporate the five verification methods described in the following subparagraphs.

4.1.2.1 Inspection. Inspection shall include formal verification of compliance with a requirement by examination of CI's or the assembled CPCI and associated design documentation at the time and place of qualification testing.

4.1.2.2 Analysis. Analysis shall include formal verification of a performance requirement by examination and study of the computer program design and coding. Verification of compliance of a requirement may be accomplished through analysis of algorithms and the flow of input data through successive stages of processing.

4.1.2.3 Demonstration. Demonstration shall include formal verification of performance characteristics through observation of functions being performed by the operating computer program. Verification shall be accomplished at the time and place of the demonstration.

4.1.2.4 Test. Test shall include formal verification of a performance requirement by exercising a specific computer program function using pre-defined parameters which will generate a specific and predictable program response or output. Verification may be accomplished by demonstration or by analysis or inspection of output displays or hard copy results.

4.1.2.5 Review of test data. Test data review shall include review of test records for test/demonstrations accomplished at an earlier time. This method is typical of requirements tested during CPT&E, but may also apply to other requirements which depend on a series of tests over more than one test occasion or under varied conditions of operation. Verification shall be accomplished by review of (a) detailed test procedures, including input data, and (b) hardcopy printouts of CPCI test outputs.

4.1.3 Test constraints. Tests shall be combined when feasible. Test data available from other sources or obtained during early stages of development testing shall be used to the maximum extent possible to reduce duplication and costs. Any component or CI which fails to meet all applicable requirements of this specification shall be considered as rejected until corrective action has been completed. A test may be continued while the extent and cause of a failure is being determined, provided the failure does not negate test results for the remainder of the test.

4.2 Computer programming test and evaluation. The developer may incorporate his own internal plans and procedures for conducting initial CPT&E. This may include code checking, debugging and preliminary performance testing. Final CPT&E verification procedures shall incorporate the following:

4.2.1 Host compiler compatibility. All government furnished program modules and all contractor developed or modified program modules shall be compiled using the host system compilers defined in 3.1.2.2.5. All compiler SYNTAX errors, greater than error level 4, shall be corrected prior to PQT or higher level tests.

4.2.2 Programming standards. CPT&E shall include verification that all program modules are in compliance with the programming methods and standards defined in 3.3.1 and subparagraphs therein.

4.3 Preliminary qualification tests. To the maximum extent possible, PQT shall incorporate test methods and procedures designed to duplicate the same processing results as those obtained by the same techniques, processes, or programs previously executed in a batch mode. Additional PQT requirements and methods of verification for Section 3 requirements are defined in the Verification Cross Reference Matrix in 4.6.

4.4 Formal qualification tests. To the maximum extent possible, FQT shall incorporate evaluation and analysis of the testing results obtained during PQT. Additional FQT requirements and methods of verification for Section 3.0 requirements are defined in the Verification Cross Reference Matrix in 4.6.

4.5 System test program. System test is not applicable to this specification.

4.6 Verification cross-reference matrix. Verification of Section 3.0 requirements shall be accomplished in accordance with the verification methods and test types defined in Table 4.6.1.

Table 4.6.1 Verification Cross Reference Index

Verification Method

1 Inspection (4.1.2.1)
2 Analysis (4.1.2.2)
3 Demonstration (4.1.2.3)
4 Test (4.1.2.4)
5 Review of Test Data (4.1.2.5)

Test Category

A CPT&E (4.1.1.1)
B PQT (4.1.1.2)
C FQT (4.1.1.3)

NOTE: The verification method, test category, and associated test requirements specified for each paragraph shall apply to all subparagraphs therein when the associated subparagraphs are not separately listed in this index.

| Section 3 Reference | Requirement | Verification Method | | | | | Test Category | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | A | B | C |
| 3.1.1.1 | Program execution area/overlay capability | X | | | | | X | | |
| 3.1.1.2 | Refresh buffer capacity | X | | | | | X | | |
| 3.1.1.3 | On-line load module storage | | | X | | | | X | X |
| 3.1.1.5 | Response to operator requests | | | X | | | | X | X |
| 3.1.1.5 | Display self-modification functions | | | X | | | | X | X |
| 3.1.1.5 | Interrupt handling during refresh | | | X | | | | X | X |
| 3.1.1.6 | Program source code backup | X | | | | | | | X |
| 3.1.2.1 | Minimum Functional interfaces | X | | | | | X | | |
| 3.1.2.1.1 | Interface Security | X | | | | | X | | |
| 3.1.2.2 | Adage 4250 graphics system | | | X | | | X | | |

Table 4.6.1 Verification Cross Reference Index (Continued)

| Section 3 Reference | Requirement | Verification Method | | | | | Test Category | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | A | B | C |
| 3.1.2.2 | IBM 4341 Host computer system | | | X | | | X | | |
| 3.1.2.2.1 | Graphics peripheral devices | X | | X | | | X | | |
| 3.1.2.2.2 | Host devices | | | X | | | X | | |
| 3.2 | Interactive program execution | | | X | | | | | X |
| 3.2.1.1 | Remote Job Entry (RJE) capability | | | X | | | | X | X |
| 3.2.1.2.1 | Program initiation | | X | X | | | | X | X |
| 3.2.1.2.2 | Primary control display (PCD) | | | X | | | | X | X |
| 3.2.1.2.2.1 | Display title | | | X | | | | X | X |
| 3.2.1.2.2.2 | Function selection | | | X | | | | X | X |
| 3.2.1.2.2.3 | Common function control | | | X | | | | X | X |
| 3.2.1.2.2.4 | Data identification control fields | | | X | | | | X | X |
| 3.2.1.2.3 | Reset feature | | X | X | | | | X | X |
| 3.2.1.2.4 | Executive control | | | X | | | | X | X |
| 3.2.1.2.5 | Menu display | | | X | | | | X | X |
| 3.2.1.2.6 | Help displays | | | X | | | | X | X |
| 3.2.1.2.7 | Help accessibility | | | X | | | | X | X |
| 3.2.1.2.8 | Program termination | | X | X | | | | X | X |
| 3.2.1.2.9 | Graphics display and buffer wipeout | | | X | | | | X | X |
| 3.2.2.2.1 | WBASE input/output | | X | X | | | | X | X |
| 3.2.2.2.2 | BSTALOCS input | | X | | | | | X | |

Table 4.6.1 Verification Cross Reference Index (Continued)

| Section 3 Reference | Requirement | Verification Method | | | | | Test Category | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | A | B | C |
| 3.2.2.2.3 | Storage of measurements | | | X | X | | | X | X |
| 3.2.2.2.4 | Archives access | | | | X | | | X | X |
| 3.2.3.2.1 | Binary gain ranging conversions | | X | X | | | | X | X |
| 3.2.3.2.2 | Directory scan | | | X | | | | X | X |
| 3.2.3.2.3 | Waveform scan and review | | | X | | | | X | X |
| 3.2.3.2.3.1 | Scan time base | | | X | | | | X | X |
| 3.2.3.2.3.2 | Automatic marking and alignment | | | X | | | | X | X |
| 3.2.3.2.3.3 | Waveform annotation | | | X | | | | X | X |
| 3.2.3.2.3.4 | Window definition | | | X | | | | X | X |
| 3.2.3.2.4 | Waveform select/delete | | | X | | | | X | X |
| 3.2.3.2.5 | Phase marking | | | X | | | | X | X |
| 3.2.3.2.6 | Scan/edit display paging | | | X | | | | X | X |
| 3.2.3.2.7 | Identification of clipped data | | | X | | | | X | X |
| 3.2.4.2.1 | Spectrum computation | | | X | X | | | X | X |
| 3.2.4.2.2 | Instrument response removal | | | | X | X | | X | X |
| 3.2.4.3 | Output corrected signal | | | | X | X | | X | X |
| 3.2.5.2.1 | Mode identification | | | X | X | | | X | X |
| 3.2.5.2.2 | Window definition | | | X | X | | | X | X |
| 3.2.5.3 | Output group velocity | | | X | X | | | X | X |
| 3.2.6.2.2 | Interpolation | | | X | X | | | X | X |

Table 4.6.1 Verification Cross Reference Index (Continued)

| Section 3 Reference | Requirement | Verification Method | | | | | Test Category | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | A | B | C |
| 3.2.6.2.3 | Phase unwrapping | | X | X | X | | | X | X |
| 3.2.6.3 | Output corrected group and phase velocity | | | X | X | | | X | X |
| 3.2.7.2.1 | Damping | | | X | X | | | X | X |
| 3.2.7.2.2 | Inversion for earth structure | | | X | X | | | X | X |
| 3.2.7.2.2 | Inversion for attenuation/scalar moment | | | X | X | | | X | X |
| 3.2.7.2.3 | Root search | | X | X | X | | | X | X |
| 3.2.7.3 | Output attenuation/earth structure | | | X | X | | | X | X |
| 3.2.8 | Synthetic seismograms | | | X | X | | | X | X |
| 3.2.8.3 | Output synthetic seismograms | | | X | X | | | X | X |
| 3.2.9.2 | Construct master filters | | | X | X | | | X | X |
| 3.2.9.3 | Output master filters | | | X | X | | | X | X |
| 3.2.10.2 | Calculate spectral magnitude | | | X | X | | | X | X |
| 3.2.10.3 | Output spectral magnitude | | | X | X | | | X | X |
| 3.2.11.2 | Calculate scalar moment | | | X | X | | | X | X |
| 3.2.11.3 | Output scalar moment | | | X | X | | | X | X |

## 5.0 PREPARATION FOR DELIVERY.

5.1 Classified material. Packaging and handling of all classified software and classified test results shall be in accordance with the packaging and handling requirements defined in the Surface Wave Magnitude Studies Security Instruction for project authorization number T/5102, 26 February 1985.

## 6.0 NOTES

## APPENDIX I

### 10.0 EXPLOSION SYNTHETIC SEISMOGRAMS.

10.1 Purpose. This appendix has been included to explain the notation convention adopted for the SWANS.

10.2 Origin. The pages have been adapted directly as received from Dr. D. Russell at St. Louis University.

## EXPLOSION SYNTHETIC SEISMOGRAMS

This study will outline the mathematical equations used for vertical component, fundamental mode surface wave seismograms, assuming an isotropic explosion source. If a step source time function is assumed, and if the receiver is at the surface, the vertical component can be expressed as (Levshin and Yanson, 1971; Herrmann, 1974; Aki and Richards, 1980):

$$u_z(\omega) \;=\; \left(\frac{M_0}{j\,\omega}\right)\left(\frac{A_r}{2}\right)\left[\frac{dr_2(h)}{dz} - k\ r_1(h)\right]\left(\frac{2}{\pi k x}\right)^{1/2} e^{-i(\mathbf{k}x\,+\,\pi/4)} \qquad (1)$$

where

$u_z(\omega)$ = vertical displacement

$M_0$ = scalar moment

$A_r$ = path response

$k$ = $\omega / c$ = wavenumber

$c$ = phase velocity

$\omega$ = angular frequency

$x$ = source/receiver distance

$\mathbf{k}$ = $k + i\gamma$ = complex wavenumber

$\gamma$ = attenuation coefficient

$r_1$ = radial displacement eigenfunction

$r_2$ = vertical displacement eigenfunction

$h$ = source depth .

Harkrider (1964,1970) gives an alternative expansion for the explosion seismogram. The above closely follows that of Levshin and Yanson, and Aki and Richards. It differs from Aki's notation in that a reverse sign convention is used for wavenumbers in Fourier transforms. Also, Aki's definition of the path response ($A_r$) is twice that defined here. This is due to his energy integrals being defined as one-half the value defined in Levshin and Yanson, Herrmann, and Harkrider.

From the stress - displacement relations for Rayleigh waves (Aki and Richards, 1980), it can be directly shown that

$$\frac{dr_2(h)}{dz} - k\ r_1(h) \ = \ \frac{2\beta^2}{\alpha^2}\left(\frac{r_4(h)}{2\rho\beta^2} - k\ r_1(h)\right) \qquad (2)$$

where

$\alpha$ = compressional velocity at the source

$\beta$ = shear velocity at the source

$\rho$ = density at the source

$r_4(h)$ = vertical stress eigenfunction.

Substituting (2) into (1) and rearranging gives

$$u_z(\omega) \ = \ M_0'\ A_r\ Q(\omega)\left(\frac{2}{9\pi\omega^2 kx}\right)^{1/2} e^{-i(\mathbf{k}x\ +\ 3\pi/4)} \qquad (3)$$

where

$$M_0' \ = \ \frac{3\beta^2}{\alpha^2} M_0 \ = \ \text{normalized moment (Stevens, 1986)}$$

$$Q(\omega) \ = \ \left(\frac{r_4(h)}{2\rho\beta^2} \ - \ k\ r_1(h)\right) .$$

Notice that the imaginary term below the scalar moment has been incorporated in the phase, and that the corresponding $\omega$ is now under the radical.

For separate source and path regions, equation (3) can be modified to (Bache *et. al.*, 1978a):

$$u_z(\omega) = M_0' A_{r1} Q_1(\omega) T(\omega) \left( \frac{2}{9\pi\omega^2(k_1x_1+k_2x_2)} \right)^{1/2} e^{-i(\mathbf{k}_1x_1+\mathbf{k}_2x_2+3\pi/4)} . \quad (4)$$

The subscript 1 indicates the source region and 2 indicates the path region. The expression

$$T(\omega) = \left( \frac{k_1 A_{r2}}{k_2 A_{r1}} \right)^{1/2}$$

is an "impedence matching" term which guarantees that the total energy flux across the source/path boundary will remain constant, assuming no mode conversion (Bache *et. al.*, 1978a).

If the source region is small in comparision with the path, $x_1$ can be considered as negligible and set to zero. Assuming this is true, and grouping all terms in (4) according to source and path, results in

$$u_z(\omega) = M_0' \; Q_1(\omega) \left( \frac{2k_1A_{r1}}{9\pi\omega^2} \right)^{1/2} \frac{(A_{r2})^{1/2}}{k_2} \frac{e^{-i(\mathbf{k}_2x_2 + 3\pi/4)}}{x_2^{1/2}} . \quad (5)$$

The distance term ($x_2^{1/2}$) under the exponent defines the geometrical spreading.

For large source/receiver distances, the sphericity of the earth must taken into account in (5). It is also convenient to express the explosion seismogram in terms of phase velocities, instead of wavenumbers. Rewriting (5) in terms of phase velocities ($c = \omega/k$) and incorporating spherical geometrical spreading (Ben-Menahem and Singh, 1981) gives

$$u_z(\omega) = M_0' \; S_1(\omega) \; S_2(\omega) \; \frac{e^{-i[(\omega/c_1 + i\gamma_1) + 3\pi/4]}}{[a_e \sin(x_2/a_e)]^{1/2}} \tag{6}$$

where $a_e$ is the radius of the earth,

$$S_1(\omega) = \left( \frac{2A_{r1}}{9\pi\omega^3 c_1} \right)^{1/2} Q_1(\omega)$$

$$S_2(\omega) = c_2 \, (A_{r2})^{1/2} \; .$$

The above is precisely the same as found by Stevens (1986), except for $S_1(\omega)$. The difference is due to Stevens using Harkrider's mathematical development. To convert to Harkrider's notation, first expand $S_1(\omega)$:

$$S_1(\omega) = \left( \frac{2A_{r1}}{9\pi\omega^3 c_1} \right)^{1/2} \left( \frac{r_4(h)}{2\rho\beta^2} - \frac{\omega}{c_1} r_1(h) \right) .$$

Factor $\omega/c_1$ from $Q_1(\omega)$ for

$$S_1(\omega) = \left( \frac{2A_{r1}}{9\pi\omega c_1^3} \right)^{1/2} \left( \frac{c_1 r_4(h)}{2\omega\rho\beta^2} - r_1(h) \right) . \tag{7}$$

Define the following equivalent expressions to Harkrider (1964,1970):

$$\begin{aligned} \frac{r_4(h)}{\omega/c_1} &= \frac{\sigma_2^*(h)}{\dot{\omega}_0/c} = E_3 \\ r_1(h) &= \dot{u}_s^*/\dot{\omega}_0 = E_1 \\ \mu &= \rho\,\beta^2 \; . \end{aligned}$$

Substitute these into (7) for

$$S_1(\omega) = \left( \frac{2A_{r1}}{9\pi\omega c_1^3} \right)^{1/2} \left( \frac{E_3}{2\mu} - E_1 \right) \tag{8}$$

which is equivalent to Stevens (1986) result.

## References

Aki, K. and P. Richards (1980). *Quantitative Seismology: Theory and Methods,* W. H. Freeman and Co.

Ben-Menahem, A. B. and S. J. Singh (1981). *Seismic Waves and Sources,* Springer-Verlag New York Inc.

Bache, T. C., W. L. Rodi, and D. G. Harkrider (1978a). Crustal structures inferred from Rayleigh wave signatures of NTS explosions, *Bull. Seism. Soc. Am. 68,* 1399-1413.

Harkrider, D. G. (1964). Surfacewaves in multilayered media. I. Rayleigh and Love waves from buried sources in a multilayered elastic half space, *Bull. Seism. Soc. Am. 54,* 627-629.

Harkrider, D. G. (1970). Surface waves in multilayered media. II. Higher mode spectra and spectral ratios from point sources in plane-layered earth models, *Bull. Seism. Soc. Am. 60,* 1937-1987.

Herrmann, R. B. (1974). *Surface Wave Generation by Central United States Earthquakes,* Ph.D. Dissertation, Saint Louis University, 263 pp.

Levshin, A. L. and A. Z. Yanson (1971). Surface waves in vertically and radially inhomogeneous media (English translation), *Computer Programs in Earthquake Seismology, Vol. 2: Surface Wave Programs,* Department of Earth and Atmospheric Sciences, Saint Louis University, A1-A35.

Stevens, J. L. (1986). Estimation of scalar moments from explosion-generated surface waves, *Bull. Seism. Soc. Am. 76,* 123-151.

Department of Earth and Atmospheric Sciences<br>
Saint Louis University<br>
P.O. Box 8099, Laclede Station<br>
St. Louis, Missouri 63156

# APPENDIX II

## 20.0 LINEAR INVERSION THEORY.

20.1 Purpose. This appendix has been included to provide the theory for the Inversion function available in the SWANS.

20.2 Origin. These papers have been adapted directly as received from Dr. D. Russell at St. Louis University.

# LINEAR INVERSION THEORY IN GEOPHYSICS

Many geophysical problems can be modeled as a Fredholm integral of the first kind (Twomey, 1977, p. 5):

$$y(t) = \int_r A(t,r)\, x(r)\, dr \qquad (1)$$

where

$t,r =$ coordinate systems for data space and model space,

$x(r) =$ model parameters desired, such as density, velocity, or Q,

$y(t) =$ observational measurements,

$A(t,r) =$ integral kernel relating model parameters to observations.

In the context of equation (1), the purpose of linear inversion is to find physically acceptable solutions to the model $x(r)$, which will in some sense satisfy the observational data $y(t)$ .

Backus and Gilbert (1967, 1968) pointed out that although geophysical models can generally be regarded as continuous functions of $r$, "observational measurements" usually imply a discrete set of data points. In this case, (1) must be modified to

$$y_i = \int_r A_i(r)\, x(r)\, dr \ , \qquad i = 1,2,...m \qquad (2)$$

where $m$ is the total number of observations. There are a variety of ways of inverting (2) for $x$, given known data $y_i$. Three well known techniques are the Backus-Gilbert method, series expansion, and model discretization. These will be briefly outlined below.

### Backus-Gilbert method

The Backus-Gilbert method (Backus and Gilbert, 1967, 1968) is in a sense the most general solution possible for (2), in that it requires no *a priori* parameterization of the model $x(r)$. The solution is continuous, and is constrained only by the maximum resolution obtainable by the particular Fredholm integral used.

The model is constructed at any continuous point $r'$ as a linear combination of all the data points $y_i$. Specifically, both sides of equation (2) are summed as follows:

$$\sum_{i=1}^{m} h_i(r')\, y_i \;=\; \sum_{i=1}^{m} h_i(r') \int_r A_i(r)\, x(r)\, dr \;, \tag{3}$$

where $h_i(r')$ is a set of as yet unknown constants evaluated at point $r'$. Equation (3) can be written as

$$\sum_{i=1}^{m} h_i(r')\, y_i \;=\; \int_r R(r',r)\, x(r)\, dr \;, \tag{4}$$

where

$$R(r',r) \;=\; \sum_{i=1}^{m} h_i(r')\, A_i(r)\,. \tag{5}$$

The essence of the Backus-Gilbert method is to find a set of $h_i(r')$ that will allow equation (5) to approximate a Dirac delta function:

$$R(r',r) \;=\; \sum_{i=1}^{m} h_i(r')\, A_i(r) \;\approx\; \delta(r'-r)\,. \tag{6}$$

Substitute (6) into (4) for

$$\int_r R(r',r)\, x(r)\, dr \;\approx\; \int_r \delta(r'-r)\, x(r)\, dr \;=\; \hat{x}(r')\,, \tag{7}$$

which gives the model estimate as a linear combination of the data

$$\hat{x}(r') = \sum_{i=1}^{m} h_i(r')\, y_i \ . \tag{8}$$

Equation (5) is the "resolution kernel" for the model estimate (8). The estimate will be a weighted average of possible models about position $r'$, as shown by equation (7).

Applying the Backus-Gilbert method requires constructing a least-squares minimization function in terms of the residuals between the resolution kernel and the Dirac delta function in (6). For each point $r'$, a set of $h_i(r')$ is found which will minimize the sum of the squares of the residuals. Additional constraints to control the variance of the solution can be incorporated into the minimization function, leading to a "tradeoff" between resolution and variance. This procedure is discussed in detail by Backus and Gilbert (1967, 1968).

An advantage of the Backus-Gilbert method is that the form of the solution depends only on the original equation (2). No parameterization of the model is required, such as "block modeling" or model discretization. A disadvantage is that the inversion coefficients $h_i(r')$ must be reconstructed in a least squares system of equations for every point $r'$. When the number of observations becomes large, the time to find a solution for many points in model space can become unacceptably long.

The other methods discussed below can be computationally more efficient than Backus-Gilbert, but at the expense of parameterizing the form of the model.

### Series expansion

For particular cases, it may be possible to parameterize the model in terms of a truncated series of linearly independent basis functions:

$$x(r) = \sum_{j=1}^{n} b_j \, f_j(r) \tag{9}$$

where

$f_j(r) =$ basis function,

$b_j =$ basis function coefficient,

$n =$ total number of coefficients.

For example, Woodhouse and Dziewonski (1984) took advantage of the sphericity of the earth and modeled the three-dimensional velocity distribution as a set of spherical harmonic basis functions.

Substituting (9) into (2) gives

$$y_i = \int_r A_i(r) \sum_{j=1}^{n} b_j \, f_j(r) \, dr$$

which can be rearranged as

$$y_i = \sum_{j=1}^{n} \{ \int_r A_i(r) f_j(r) dr \} \, b_j \, . \tag{10}$$

Considering the integral in brackets as elements of a matrix, (10) can be written in the simple form

$$y_i = \sum_{j=1}^{n} A_{ij} \, b_j \, . \tag{11}$$

Equation (11) can be inverted for $b_j$ by standard least-squares algorithms. Once the coefficients have been found, they can be substituted into equation (9) to form a continuous solution.

### Model discretization

This method is applicable in cases where the model can be approximated by discrete, or piece-wise constant values over the model space. With no loss of generality, equation (2) can be written as a sum of integrals over subregions of the model space:

$$y_i = \sum_{j=1}^{n} \int_{r_j} A_i(r)\, x(r)\, dr \ . \tag{12}$$

The subscript $r_j$ represents the integral boundaries on the j'th subregion. If $x(r)$ is constant within each subregion, it can be factored from the integrals in (12) for

$$y_i = \sum_{j=1}^{n} \{ \int_{r_j} A_i(r)\, dr \}\, x_j \ , \tag{13}$$

where $x_j$ is the constant value of $x(r)$ in the subregion. This can be simplified to

$$y_i = \sum_{j=1}^{n} A_{ij}\, x_j \ , \tag{14}$$

where $A_{ij}$ represents the integral in brackets.

Model discretization is frequently used for tomography problems (McMechan, 1983), where the velocity structure is subdivided into a large number of blocks, with a constant velocity assumed in each block. The forward problem (12) is the observed travel time as an integral function of wave slowness across the structure. The method is also used for problems that assume a spherical or plane layered earth structure. "Plane layered structure" is defined as a medium which is laterally homogeneous, with a piece-wise constant depth variation. It is particularly applicable for plane layered surface wave analysis, since the integral in (13) can be analytically evaluated.

All the above methods result in matrix equations similar to (14). Further analysis requires finding an inverse solution to (14) in terms of $x_j$. This can be complicated by instability in the system, occurring when small errors in $y_i$ are magnified into large errors in $x_j$. The mathematical procedure of finding stable solutions to (14) will be discussed below in terms of orthogonal decompositions.

### Least squares analysis and the singular value decomposition

Equation (14) assumes that the observed data will exactly equal the theoretical values, which is not true in practice. To compensate for errors, a residual term is added to (14):

$$y_i = \sum_{j=1}^{n} A_{ij} \, x_j \; + \; \epsilon_i \; . \qquad (15)$$

It is convenient at this point to write (15) in vector-matrix form, so the entire set of observed data can be represented in one equation. Let

$$\mathbf{y} = \mathbf{A}\,\mathbf{x} + \boldsymbol{\epsilon} \, , \qquad (16)$$

where

$\mathbf{y} =$ $m$ x 1 vector of observations,

$\mathbf{x} =$ $n$ x 1 vector of unknown parameters,

$\boldsymbol{\epsilon} =$ $m$ x 1 vector of residuals,

$\mathbf{A} =$ $m$ x $n$ kernel matrix.

The convention used is that any bold-faced entry indicates a vector or matrix, while all other variables are scalar.

The principle of least squares states that a valid set of model parameters ($\mathbf{x}$) is one which minimizes the sum of the squares of the residuals ($\boldsymbol{\epsilon}$). The minimization function can be represented mathematically as

$$M(\mathbf{x}) = \boldsymbol{\epsilon}^T \boldsymbol{\epsilon} = |\boldsymbol{\epsilon}|^2 = \sum_{i=1}^{m} \epsilon_i^2 \ . \tag{17}$$

The terms on the right are the same, being three different ways of expressing the squared norm of the residuals. The superscript "$T$" indicates the transpose of $\boldsymbol{\epsilon}$, and the bars "$| \, |$" indicate the root mean square norm of $\boldsymbol{\epsilon}$.

Finding a minimum for $M$ is simplified by the singular value decomposition (SVD), an orthogonal transformation of the $\mathbf{A}$ matrix. Lawson and Hanson (1974, p. 107) show that any arbitrary matrix can be transformed into

$$\mathbf{A} = \mathbf{U \Lambda V}^T \tag{18}$$

where

$\mathbf{U} =$ $m$ x $m$ orthogonal matrix,

$\mathbf{V} =$ $n$ x $n$ orthogonal matrix,

$\mathbf{\Lambda} =$ $m$ x $n$ upper left diagonal matrix.

"Orthogonal" means that the transpose of the matrix is also the inverse, and "upper left diagonal" means that $\mathbf{\Lambda}$ can be written as

$$\mathbf{\Lambda} = \begin{bmatrix} \mathbf{\Lambda}_k & 0 \\ 0 & 0 \end{bmatrix}, \tag{19}$$

where

$\mathbf{\Lambda}_k = k$ x $k$ diagonal matrix.

The terms along the diagonal are the singular values of the **A** matrix, and the rank of the **A** matrix is $k$, the number of non-zero singular values. From (19), the rank of the matrix obeys the inequality

$$k \leq min(n,m) . \tag{20}$$

Equations (18), (19), and (20) lead to a general classification for arbitrary matrices (Lawson and Hanson, 1974, p. 3). If

$n > m$ the matrix is overdetermined,

$n < m$ the matrix is underdetermined,

$n = m$ the matrix is even determined,

$k = min(n,m)$ the matrix has full rank,

$k < min(n,m)$ the matrix is rank deficient, or underconstrained.

The matrix may have full rank, but one or more of the singular values may be much less than the maximum. Given $\lambda_i$ as any singular value of $\mathbf{\Lambda}$, if $\lambda_i << \lambda_{\mathrm{max}}$ , the matrix is poorly constrained. This condition is common in geophysical problems.

To transform the least squares problem, substitute (18) into (16) for

$$\mathbf{y} = \mathbf{U \Lambda V}^T \mathbf{x} + \boldsymbol{\epsilon} .$$

Multiply both sides by $\mathbf{U}^T$, the inverse of **U** for

$$\mathbf{U}^T \mathbf{y} = \mathbf{\Lambda}\, \mathbf{V}^T \mathbf{x} + \mathbf{U}^T \boldsymbol{\epsilon} .$$

Make the following variable substitutions

$$\mathbf{g} = \mathbf{U}^T\mathbf{y}\ , \quad \mathbf{p} = \mathbf{V}^T\mathbf{x}\ , \quad \mathbf{e} = \mathbf{U}^T\boldsymbol{\epsilon} \tag{21}$$

for

$$\mathbf{g} \;=\; \boldsymbol{\Lambda}\mathbf{p} \;+\; \mathbf{e}\ . \tag{22}$$

Equation (22) is equivalent to the original least squares problem since the minimization function for (22) is the same as (17):

$$M(\mathbf{p}) \;=\; \mathbf{e}^T\mathbf{e} \;=\; \boldsymbol{\epsilon}^T\mathbf{U}\mathbf{U}^T\boldsymbol{\epsilon} \;=\; \boldsymbol{\epsilon}^T\boldsymbol{\epsilon}\ . \tag{23}$$

This demonstrates that the least squares problem is invariant to multiplication by orthogonal matrices.

To find the least squares solution, partition (22) according to the rank of **A**:

$$\begin{bmatrix}\mathbf{g}_k \\ \mathbf{g}_{m-k}\end{bmatrix} = \begin{bmatrix}\boldsymbol{\Lambda}_k & 0 \\ \mathbf{0} & 0\end{bmatrix}\begin{bmatrix}\mathbf{p}_k \\ \mathbf{p}_{n-k}\end{bmatrix} + \begin{bmatrix}\mathbf{e}_k \\ \mathbf{e}_{m-k}\end{bmatrix}\ . \tag{24}$$

The subscripts indicate the dimensions of the vectors and matrix. From (23) and (24), the minimization function is

$$M(\mathbf{p}) \;=\; |\mathbf{e}\,|^2 \;=\; |\mathbf{e}_k\,|^2 \;+\; |\mathbf{e}_{m-k}\,|^2\ ,$$

which from (24) is

$$M(\mathbf{p}) \;=\; |\mathbf{g}_k - \boldsymbol{\Lambda}_k\mathbf{p}_k\,|^2 \;+\; |\mathbf{g}_{m-k}\,|^2\ . \tag{25}$$

The minimum of (25) corresponds to the vector $\mathbf{p}_k$ which sets the first norm to zero, and this can be found immediately from

$$\mathbf{p}_k \;=\; \boldsymbol{\Lambda}_k^{-1}\,\mathbf{g}_k\ . \tag{26}$$

From (26), (24), and (21), the solution to the least squares problem is

$$\mathbf{p} \;=\; \begin{bmatrix}\mathbf{p}_k \\ \mathbf{p}_{n-k}\end{bmatrix}, \qquad \mathbf{x} \;=\; \mathbf{V}\mathbf{p}\ . \tag{27}$$

Equation (25) is remarkable in that the least squares solution can be found from algebraic considerations alone, without resorting to differential calculus. The solution is completely general; no restrictions are made on the dimensions or rank of **A**. Another feature of (25) is that the minimum residual can be determined directly from $\mathbf{g}_{m-k}$. Notice that for full rank underdetermined problems, $m = k$ and from (25), the minimum residual is zero.

Unless the problem is full rank and overdetermined $(n = k)$, the solution will be non-unique. Only $\mathbf{p}_k$ is necessary to minimize the residual in (25), and inspection of (24) shows that $\mathbf{p}_{n-k}$ is arbitrary. Therefore, there are an infinite number of possible values for **p** in (27). This leads to the concept of the "minimum length" solution for underdetermined (or underconstrained) problems. If $\mathbf{p}_{n-k}$ is arbitrarily set to zero, a valid least squares solution is, from (27)

$$\mathbf{x} = \mathbf{V}\begin{bmatrix}\mathbf{p}_k \\ 0\end{bmatrix}. \tag{28}$$

Substituting (26) into the minimum length solution (28), and transforming back to the original variables in (21) gives

$$\mathbf{x} = \mathbf{V}\mathbf{\Lambda}^{-1}\,\mathbf{U}^T\mathbf{y} \tag{29}$$

where

$$\mathbf{\Lambda}^{-1} = \begin{bmatrix}\mathbf{\Lambda}_k^{-1} & 0 \\ 0 & 0\end{bmatrix}.$$

The dimensions of $\mathbf{\Lambda}^{-1}$ are $n$ x $m$. This leads to the definition of the generalized inverse (Aki and Richards, 1980, p.684):

$$\mathbf{H}_g = \mathbf{V}\mathbf{\Lambda}^{-1}\mathbf{U}^T . \tag{30}$$

For full rank problems, there are two equivalent forms to the generalized inverse. If the problem is overdetermined, the inverse can be written

$$\mathbf{H}_o = (\mathbf{A}^T\mathbf{A})^{-1}\,\mathbf{A}^T \tag{31}$$

and for underdetermined problems, the inverse is

$$\mathbf{H}_u = \mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}\,. \tag{32}$$

These can be verified by substituting the singular value decomposition (18) into (31) and (32). Under the stated constraints (full rank, over or underdetermined), multiplying out the matrices shows that the two forms are exactly equal to the generalized inverse (30). For even determined systems, (31) and (32) are equivalent.

Another way of expressing the generalized inverse is in terms of a vector sum. Letting each column of **U** and **V** represent orthogonal vectors $\mathbf{u}_j$ and $\mathbf{v}_j$, matrix manipulation of (29) leads directly to the sum

$$\mathbf{x} = \sum_{j=1}^{k} \frac{\mathbf{u}_j^T\mathbf{y}}{\lambda_j}\,\mathbf{v}_j \tag{33}$$

This gives the solution vector as a sum of $k$ orthogonal vectors $\mathbf{v}_j$. For poorly constrained problems, at least one of the singular values ($\lambda_j$) will be small, and this can lead to excessive magnification of the corresponding vector $\mathbf{v}_j$. This defines an unstable least squares problem, and the next section will discuss inverse solutions which control the instability.

### Stochastic inversion

Generalized inversion gives solutions to rank deficient least squares problems, but it does not address poorly constrained problems, where singular values approach, but do not exactly equal zero. One method of dealing with small singular values is to set them equal to zero, reducing the rank of the problem. This is the "sharp cutoff" technique (Wiggins, 1972). In (33), only $\mathbf{v}_j$ vectors with relatively large singular values are included in the sum, in order to produce physically reasonable solutions. In practice, however, this method sometimes leads to unwanted "ripples" or oscillations in the solution, due to the abrupt truncation of the vector sum (33). The effect is similar to the problem of using ideal low-pass filters in Fourier analysis.

Another approach to instability is to constrain the norm of the solution vector. In (33), small singular values can produce solutions with large magnitudes. To control this, the Levenburg-Marquardt damped least squares method (Levenburg, 1944; Marquardt, 1963) includes the norm of the solution vector as a part of the least squares minimization function. This is implemented by appending a scaled identity matrix to the original least squares problem:

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{A} \\ \gamma \mathbf{I} \end{bmatrix} \mathbf{x} + \begin{bmatrix} \boldsymbol{\epsilon} \\ \boldsymbol{\epsilon}_\gamma \end{bmatrix} . \tag{34}$$

The scalar variable $\gamma$ is undetermined at this point. If it is set equal to zero, (34) reduces to the original least squares problem. The minimization function is

$$M(\mathbf{x}) = |\boldsymbol{\epsilon}|^2 + |\boldsymbol{\epsilon}_\gamma|^2$$

which from (34) is

$$M(\mathbf{x}) = |\mathbf{y} - \mathbf{A}\mathbf{x}|^2 + \gamma^2 |\mathbf{x}|^2 . \tag{35}$$

As the value of $\gamma^2$ increases, more weight is put on minimizing the solution norm, and less on the least squares residual. This insures stable solution vectors, but at the expense of larger least squares residuals.

No matter what the rank and dimension of the original problem, equation (34) is overdetermined and full rank. This is obvious since the scaled identity matrix is even determined and full rank. It can be expressed as an independent least squares problem

$$\hat{\mathbf{y}} = \hat{\mathbf{A}}\mathbf{x} + \hat{\boldsymbol{\epsilon}} ,$$

where the new variables are equivalent to the partitioned ones in (34). Since this problem is full rank and overdetermined, equation (31) is a valid inverse:

$$\mathbf{x} = (\hat{\mathbf{A}}^T\hat{\mathbf{A}})^{-1}\, \hat{\mathbf{A}}^T\, \hat{\mathbf{y}} .$$

Substituting back the original variables in (34) and multiplying out the partitioned matrices yields

$$\mathbf{x} = (\mathbf{A}^T\mathbf{A} + \gamma^2\mathbf{I})^{-1}\, \mathbf{A}^T\, \mathbf{y} .$$

This defines the Levenburg-Marquardt damped least squares inverse

$$\mathbf{H}_d = (\mathbf{A}^T\mathbf{A} + \gamma^2\mathbf{I})^{-1}\, \mathbf{A}^T . \qquad (36)$$

For non-zero $\gamma$, $\mathbf{H}_d$ is completely general in that no restrictions on rank or dimensions are necessary for the $\mathbf{A}$ matrix. If $\gamma = 0$, (36) reduces to (31), and the inverse exists only if $\mathbf{A}$ is full rank and overdetermined.

The orthogonal equivalent of (34) is found by substituting the singular value decomposition (18) into (34) and multiplying both sides by the partitioned orthogonal matrix

$$\begin{bmatrix} \mathbf{U}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{V}^T \end{bmatrix} .$$

Using the variable substitutions in (21), the orthogonally transformed system is written

$$\begin{bmatrix} \mathbf{g} \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{\Lambda} \\ \gamma \mathbf{I} \end{bmatrix} \mathbf{p} + \begin{bmatrix} \mathbf{e} \\ \mathbf{e}_\gamma \end{bmatrix} . \tag{37}$$

This is equivalent to (34) since the minimization function is invariant to multiplication by orthogonal matrices (see (23)). Considering (37) as an independent least squares problem, it is full rank and overdetermined, so (31) is a valid inverse. Following the same steps as the Levenburg-Marquardt inverse results in a solution

$$\mathbf{p} = (\mathbf{\Lambda}^T \mathbf{\Lambda} + \gamma^2 \mathbf{I})^{-1} \mathbf{\Lambda}^T \mathbf{g} .$$

The inverse matrices are diagonal, so multiplying out the terms gives

$$\mathbf{p} = \tilde{\mathbf{\Lambda}}^{-1} \mathbf{g} = \begin{bmatrix} \tilde{\mathbf{\Lambda}}_k^{-1} & 0 \\ 0 & 0 \end{bmatrix} \mathbf{g} . \tag{38}$$

where $\tilde{\mathbf{\Lambda}}$ has the same form as in the generalized inverse (29), but the individual singular values on the diagonal are modified as

$$\tilde{\lambda}_i = \lambda_i + \frac{\gamma^2}{\lambda_i} . \tag{39}$$

Substituting back the original variables in (21) gives

$$\mathbf{x} = \mathbf{V} \tilde{\mathbf{\Lambda}}^{-1} \mathbf{U}^T \mathbf{y} , \tag{40}$$

which leads to the definition of the stochastic inverse

$$\mathbf{H}_s = \mathbf{V} \tilde{\mathbf{\Lambda}}^{-1} \mathbf{U}^T . \tag{41}$$

It has exactly the same form as the generalized inverse, except that the singular values are modified as in (39). The effect on the solution can be seen by modifying the equivalent solution vector sum (33)

$$\mathbf{x} = \sum_{j=1}^{k} \frac{\mathbf{u}_j^T \mathbf{y}}{(\lambda_j + \gamma^2/\lambda_j)} \mathbf{v}_j \ . \tag{42}$$

This stabilizes poorly constrained problems by increasing the size of small singular values. The value of $\gamma^2$ can be varied by trial and error in (42) to produce reasonable physical models.

The drawback of (42) is increased minimum residuals for the original least squares problem. From (38)

$$\mathbf{p}_k = \tilde{\Lambda}_k^{-1} \mathbf{g}_k \ .$$

Putting this in the original least squares minimization function (25) gives

$$M(\mathbf{p}) = |\mathbf{g}_k - \Lambda_k \tilde{\Lambda}_k^{-1} \mathbf{g}_k|^2 + |\mathbf{g}_{m-k}|^2 \ .$$

Writing this in summation form and rearranging terms results in

$$M(\mathbf{p}) = \sum_{i=1}^{k} \left[ \frac{\gamma^2}{\lambda^2 + \gamma^2} \right]^2 g_i^{\,2} + \sum_{i=k+1}^{m} g_i^{\,2} \ . \tag{43}$$

The first norm will be zero when $\gamma$ is zero, leaving the original least squares residual. As $\gamma$ increases, the total residual also increases.

The term "stochastic" comes from an independent formulation of the inverse problem based on the statistics of the data and model covariance matrices (Jordan and Franklin, 1971; Aki and Richards, 1980, p.695). A special case of this inverse is

$$\mathbf{H}_c = \mathbf{A}^T(\mathbf{A}\mathbf{A}^T + \gamma^2\mathbf{I})^{-1} \ . \tag{44}$$

Substituting the singular value decomposition (18) into (44), and multiplying out the matrices shows that (44) is equal to the orthogonal inverse (41). As long as $\gamma$ is not zero, the **A** matrix in (44) can be any rank and dimension.

Although the development is different for the above inverses, it should be emphasized that mathematically, (34), (41) and (44) are exactly equivalent for non-zero $\gamma$. Therefore, it is convenient to label all of them as "stochastic", unless there is a specific reason to differentiate between the methods.

## Filtered inversion

The Levenberg-Marquardt method of appending to the original least squares problem can be generalized to the following form:

$$\begin{bmatrix} \mathbf{y} \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{A} \\ \gamma \mathbf{F} \end{bmatrix} \mathbf{x} + \begin{bmatrix} \epsilon \\ \epsilon_\gamma \end{bmatrix} . \tag{45}$$

where

$\mathbf{F} =$ $n$ x $n$ arbitrary matrix.

The matrix can be constructed to constrain individual elements of $\mathbf{x}$ in any desired manner. This differs from the stochastic inverse, which from (34) gives equal weight to each element of $\mathbf{x}$. If $\mathbf{F}$ has an inverse, (45) can be transformed into a stochastic problem by the following operation:

$$\begin{bmatrix} \mathbf{A} \\ \gamma \mathbf{F} \end{bmatrix} \mathbf{x} = \begin{bmatrix} \mathbf{A} \\ \gamma \mathbf{F} \end{bmatrix} (\mathbf{F}^{-1}\mathbf{F})\, \mathbf{x} = \begin{bmatrix} \mathbf{A}\mathbf{F}^{-1} \\ \gamma \mathbf{I} \end{bmatrix} \mathbf{F}\mathbf{x} .$$

Make the variable substitutions

$$\hat{\mathbf{x}} = \mathbf{F}\mathbf{x} , \qquad \hat{\mathbf{A}} = \mathbf{A}\mathbf{F}^{-1} \tag{46}$$

for

$$\begin{bmatrix} \mathbf{y} \\ 0 \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{A}} \\ \gamma \mathbf{I} \end{bmatrix} \hat{\mathbf{x}} + \begin{bmatrix} \epsilon \\ \epsilon_\gamma \end{bmatrix} . \tag{47}$$

This is an equivalent stochastic problem, operating on a "filtered" set of the unknowns, $\hat{\mathbf{x}}$. From the previous section, define the stochastic solution of the filtered model as

$$\hat{\mathbf{x}} = \hat{\mathbf{H}}\,\mathbf{y}\ .$$

Transform back to the original variables (46) for

$$\mathbf{x} = \mathbf{F}^{-1}\hat{\mathbf{H}}\,\mathbf{y}\ . \tag{48}$$

From (18) and (46) let

$$\mathbf{A}\mathbf{F}^{-1} = \hat{\mathbf{A}} = \hat{\mathbf{U}}\hat{\boldsymbol{\Lambda}}\hat{\mathbf{V}}^{T}\ .$$

Substituting into (48) the stochastic inverses (36), (41), and (44) gives equivalent explicit forms for the filtered inverse

$$\mathbf{H}_{fd} = \mathbf{F}^{-1}\,(\hat{\mathbf{A}}^{T}\hat{\mathbf{A}} + \gamma^{2}\mathbf{I})^{-1}\,\hat{\mathbf{A}}^{T} \tag{49}$$

$$\mathbf{H}_{fc} = \mathbf{F}^{-1}\,\hat{\mathbf{A}}^{T}(\hat{\mathbf{A}}\hat{\mathbf{A}}^{T} + \gamma^{2}\mathbf{I})^{-1} \tag{50}$$

$$\mathbf{H}_{fs} = \mathbf{F}^{-1}\,\hat{\mathbf{V}}\,\check{\hat{\boldsymbol{\Lambda}}}^{-1}\,\hat{\mathbf{U}}^{T} \tag{51}$$

The filtered inverse is useful in problems where a direct minimization of the solution vector norm is inappropriate. Instead, it puts constraints on the norm of a linear transform of the solution vector ($\hat{\mathbf{x}}$). The next section will discuss a specific example of this, which is useful in one-dimensional problems.

### Differential inversion

In one-dimensional problems involving gradients, such as the velocity distribution in plane-layered media, a natural constraint is on the norm of the solution gradient, instead of the solution magnitude. This can be realized by a first-order difference filter operating on the solution vector.

Let

$$\mathbf{x}' = \mathbf{F}' \, \mathbf{x} \, ,$$

where

$$\mathbf{F}' = \begin{bmatrix} 1 & -1 & 0 & 0 & & & & 0 \\ 0 & 1 & -1 & 0 & . & . & . & 0 \\ 0 & 0 & 1 & -1 & & & & 0 \\ & . & & & . & & & 0 \\ & . & & & & . & & 0 \\ & . & & & & & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} . \tag{52}$$

If $z$ is the one-dimensional coordinate, then $\mathbf{x}'$ is the approximate differential of $\mathbf{x}$ at $z$. The inverse to $\mathbf{F}'$ exists, being an integration operator defined by the upper triangular matrix

$$\mathbf{F}'^{\,-1} = \begin{bmatrix} 1 & 1 & 1 & 1 & & & 1 \\ 0 & 1 & 1 & 1 & . & . & 1 \\ 0 & 0 & 1 & 1 & & & 1 \\ 0 & 0 & 0 & 1 & & & 1 \\ & . & & & . & & 1 \\ & . & & & & . & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} .$$

Equation (52) defines a linear transform of $\mathbf{x}$, so the filtered inverse of the previous section applies. From (46) and (47), the least squares minimization function is

$$M(\mathbf{x}) = |\mathbf{y} - \mathbf{A}\mathbf{x}\,|^2 + \gamma^2\,|\mathbf{x}'\,|^2 \, .$$

The constraint for stabilizing the least squares problem is now in terms of the differential $\mathbf{x}'$. The solution vector can be found by appropriate inverse given by (49 - 51).

## Data and model covariance

In order to completely describe observed data, it is necessary to know the error associated with each data point. This is accomplished by treating the observations as random variables, with known means and covariances. The model can also be treated as a set of random variables, with means and covariances determined as functions of the observations and the inverse operators.

Jenkins and Watts (1968, p. 72) define means and covariances in terms of expected values. Given a random variable $x$, the mean is

$$<x> \;=\; \int_{-\infty}^{\infty} x\, f(x)\, dx \;, \tag{53}$$

where $f(x)$ is the probability density function of $x$. For two random variables $x_1$ and $x_2$, the covariance between them is defined as

$$<(x_1 - <x_1>)(x_2 - <x_2>)> \;=$$

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (x_1 - <x_1>)(x_2 - <x_2>)\, f_{12}(x_1,\, x_2) dx_1 dx_2 \;, \tag{54}$$

where $f_{12}(x_1, x_2)$ is the joint probability density function. The variance of a random variable is a special case of (54):

$$<(x - <x>)^2> \;=\; \int_{-\infty}^{\infty} (x - <x>)^2\, f(x)\, dx \;. \tag{55}$$

By definition, if the variables are independent, the joint probability density function is a multiple of the individual probability density functions:

$$f_{12}(x_1,\, x_2) \;=\; f_1(x_1)\, f_2(x_2) \;.$$

In this case , substituting into (54) and manipulating the integral gives

$$<(x_1 - <x_1>)(x_2 - <x_2>)> \; = \; <x_1 - <x_1>><x_2 - <x_2>> \; = \; 0 \, .$$

For a set of observations defined by the vector **y**, the vector mean is defined as the mean of each element:

$$<\mathbf{y}> \; = \; (<y_1>, <y_2>, \ldots <y_m>)^T \tag{56}$$

The transpose indicates that **y** is a column vector. The covariance is defined as the matrix

$$\mathbf{D} \; = \; < (\mathbf{y} - <\mathbf{y}>)(\mathbf{y} - <\mathbf{y}>)^T > \; . \tag{57}$$

It is understood that the expected value operates on each element of the matrix, as in (56). An important feature of the covariance matrix is symmetry:

$$\mathbf{D} \; = \; \mathbf{D}^T \, . \tag{58}$$

A feature of the expected value is linearity. Given two constant matrices **A** and **B**,

$$< \mathbf{Ay} + \mathbf{By} > \; = \; \mathbf{A} <\mathbf{y}> \; + \; \mathbf{B} <\mathbf{y}> . \tag{59}$$

This can be verified from (53) and (56).

To find the covariance matrix for the model, first assume that a least squares inverse matrix has been found relating **x** to **y** :

$$\mathbf{x} \; = \; \mathbf{H}\,\mathbf{y} \; . \tag{60}$$

If **x** and **y** are considered to be random variables, the mean of (60) is

$$<\mathbf{x}> \; = \; <\mathbf{Hy}> \; = \; \mathbf{H}<\mathbf{y}> \; . \tag{61}$$

From (57), define the model covariance as

$$\mathbf{C} \;=\; <(\mathbf{x} - <\mathbf{x}>)(\mathbf{x} - <\mathbf{x}>)^T> \;. \tag{62}$$

Substitute (60) and (61) into (62)

$$\mathbf{C} \;=\; <(\mathbf{Hy} - \mathbf{H}<\mathbf{y}>)(\mathbf{Hy} - \mathbf{H}<\mathbf{y}>)^T> \;.$$

Factor out the inverse matrix

$$\mathbf{C} \;=\; <\mathbf{H}(\mathbf{y} - <\mathbf{y}>)(\mathbf{y} - <\mathbf{y}>)^T\mathbf{H}^T> \;.$$

Use the linearity of the expected value operator for

$$\mathbf{C} \;=\; \mathbf{H} <(\mathbf{y} - <\mathbf{y}>)(\mathbf{y} - <\mathbf{y}>)^T> \mathbf{H}^T \;,$$

and substitute in (57) for the final result

$$\mathbf{C} \;=\; \mathbf{H}\,\mathbf{D}\,\mathbf{H}^T \;. \tag{63}$$

As a special case. assume that the data are independent random variables, and that the variance for all the observations are equal. Under these conditions,

$$\mathbf{D} \;=\; \sigma^2\,\mathbf{I} \;, \tag{64}$$

where $\sigma^2$ is the data variance, and **I** is an identity matrix. Substituting into (64) gives

$$\mathbf{C} \;=\; \sigma^2\,\mathbf{H}\,\mathbf{H}^T \;. \tag{65}$$

It must be emphasized at this point that (63) is an intermediate result, valid only for the special case (65). For the general data covariance matrix, the least squares minimization function should be modified to give more weight to observations with small variances, and less weight to large variances. This results in "maximum likelihood" inversion

(Menke, 1984, p. 79), which is discussed in the next section.

### Maximum likelihood inversion

Maximum likelihood is a statistical method for estimating parameters of a given probability density function, based on samples of random variables. Let

$$P(\mathbf{y}) = f(\mathbf{y}, \theta)$$

be an assumed joint probability density function of $\mathbf{y}$, where $\theta$ represents known parameters of the function, such as the variance or mean. If $\mathbf{y}_0$ is a sample of the random variable $\mathbf{y}$, the likelihood function is defined as (Jenkins and Watts, 1968, p. 116):

$$L(\theta) = f(\mathbf{y}_0, \theta) .$$

The likelihood function differs from the probability density function in that the parameters ($\theta$) are now assumed to be unknown, and the sampled values of the random variable are fixed. The maximum likelihood estimate is defined from

$$L(\hat{\theta}) = \max[\, f(\mathbf{y}_0, \theta)\,] \tag{66}$$

The estimate $\hat{\theta}$ is defined at the point which maximizes the probability of the sampled data $\mathbf{y}_0$.

To determine the mean of sampled observations, the maximum likelihood estimate is written

$$L(<\hat{\mathbf{y}}>) = \max[\, f(\mathbf{y}_0, <\mathbf{y}>)\,] \ . \tag{67}$$

Assuming that the data is related to the model by (16)

$$\mathbf{y} = \mathbf{A}\,\mathbf{x} + \epsilon \ ,$$

the mean value is

$$<\mathbf{y}> \;=\; \mathbf{A}<\mathbf{x}> + <\boldsymbol{\epsilon}> \;.$$

If it is also assumed that

$$<\boldsymbol{\epsilon}> \;=\; 0 \tag{68}$$

then $\mathbf{A}<\mathbf{x}>$ is an unbiased estimate of the mean $<\mathbf{y}>$. Substituting into (67) gives the maximum likelihood estimate in terms of the model

$$L(<\hat{\mathbf{x}}>) \;=\; \max[\, f(\mathbf{y}_0, \mathbf{A}<\mathbf{x}>)\, ] \;. \tag{69}$$

Equation (69) is the basis for maximum likelihood inversion. While least squares inversion finds a model by minimizing residuals, maximum likelihood determines the model by maximizing the probability of sampled data.

For many geophysical problems, a reasonable probability density function for observed data is the multivariate normal distribution (Menke, 1984, p. 30):

$$f(\mathbf{y}) \;=\; \frac{1}{(2\pi)^{m/2}\,|\mathbf{D}\,|^{1/2}}\exp[\,-\frac{1}{2}(\mathbf{y} - <\mathbf{y}>)^T\mathbf{D}^{-1}(\mathbf{y} - <\mathbf{y}>)\,] \tag{70}$$

where $m$ is the number of observations, and $\mathbf{D}$ is the data covariance matrix. Assuming that the model gives an unbiased estimate to the data (68), and that $\mathbf{y}_0$ is the sampled data, the maximum likelihood estimate of the model (69) is found from

$$L(<\hat{\mathbf{x}}>) \;=\; \max\left\{\exp[\,-\frac{1}{2}(\mathbf{y}_0 - \mathbf{A}<\mathbf{x}>)^T\mathbf{D}^{-1}(\mathbf{y}_0 - \mathbf{A}<\mathbf{x}>)]\right\} . \tag{71}$$

Since the argument of the exponent is negative, (71) will have a maximum when the argument is minimized. This leads to the definition of the generalized least squares minimization function:

$$M(\mathbf{x}) \;=\; (\mathbf{y} - \mathbf{Ax})^T \mathbf{D}^{-1}(\mathbf{y} - \mathbf{Ax}) \;=\; \boldsymbol{\epsilon}^T \mathbf{D}^{-1} \boldsymbol{\epsilon} \;\;, \tag{72}$$

where $\mathbf{y}$ now represents the sampled data, and $\mathbf{x}$ is the model mean.

Equation (72) can be transformed into a standard minimization function by modifying the original least squares problem (16). Since $\mathbf{D}$ is symmetric (58), $\mathbf{D}^{-1}$ is symmetric, and a matrix $\mathbf{E}$ can be found such that

$$\mathbf{D}^{-1} \;=\; \mathbf{E}^T\mathbf{E} \;. \tag{73}$$

This can be verified by writing $\mathbf{D}$ in terms of a symmetric singular value decomposition (Jackson, 1972)

$$\mathbf{D} \;=\; \mathbf{V\Lambda V}^T \;, \qquad \mathbf{D}^{-1} \;=\; \mathbf{V\Lambda}^{-1}\mathbf{V}^T \;. \tag{74}$$

Define $\mathbf{E}$ as

$$\mathbf{E} \;=\; \mathbf{\Lambda}^{-1/2}\, \mathbf{V}^T \;\;. \tag{75}$$

Then

$$\mathbf{E}^T\mathbf{E} \;=\; \mathbf{V\Lambda}^{-1/2}\, \mathbf{\Lambda}^{-1/2}\mathbf{V}^T \;=\; \mathbf{D}^{-1} \;\;,$$

and (73) is proved. To transform the original least squares problem, multiply both sides of (16) by $\mathbf{E}$:

$$\mathbf{Ey} \;=\; \mathbf{EAx} \;+\; \mathbf{E}\boldsymbol{\epsilon} \;. \tag{76}$$

This defines a new least squares problem

$$\overline{\mathbf{y}} \;=\; \overline{\mathbf{A}}\mathbf{x} \;+\; \overline{\boldsymbol{\epsilon}} \,, \tag{77}$$

with a minimization function given by

$$M(\mathbf{x}) \;=\; \overline{\boldsymbol{\epsilon}}^T \overline{\boldsymbol{\epsilon}} \;=\; \boldsymbol{\epsilon}^T\mathbf{E}^T\mathbf{E}\boldsymbol{\epsilon} \;=\; \boldsymbol{\epsilon}^T\mathbf{D}^{-1}\boldsymbol{\epsilon} \,, \tag{78}$$

which is equivalent to (72).

The above shows that a least squares problem can be transformed into a maximum likelihood problem by modifying the equations according to (76). Since this defines a new least squares problem, the inverse matrices (34), (41), (44), or (49-51) can be used to find the model estimate:

$$\mathbf{x} = \overline{\overline{\mathbf{H}}}\,\overline{\mathbf{y}} \,. \tag{79}$$

The bar above the inverse indicates that it is constructed for the transformed system (77). The model covariance matrix can be found from (63)

$$\overline{\mathbf{C}} = \overline{\overline{\mathbf{H}}}\,\overline{\mathbf{D}}\,\overline{\overline{\mathbf{H}}}^T \quad , \tag{80}$$

where $\overline{\mathbf{D}}$ is the transformed data covariance matrix, defined from (57) as

$$\overline{\mathbf{D}} = < (\overline{\mathbf{y}} - <\overline{\mathbf{y}}>)(\overline{\mathbf{y}} - <\overline{\mathbf{y}}>)^T > \,.$$

Going back to the original coordinates gives

$$\overline{\mathbf{D}} = < (\mathbf{Ey} - \mathbf{E}<\mathbf{y}>)(\mathbf{Ey} - \mathbf{E}<\mathbf{y}>)^T > \,.$$

Factoring out $\mathbf{E}$ results in

$$\overline{\mathbf{D}} = \mathbf{E} <(\mathbf{y} - <\mathbf{y}>)(\mathbf{y} - <\mathbf{y}>)^T> \mathbf{E}^T \quad ,$$

which from (57) is

$$\overline{\mathbf{D}} = \mathbf{EDE}^T \,. \tag{81}$$

From (74) and (75), the matrix in (81) can be written

$$\mathbf{EDE}^T = \mathbf{\Lambda}^{-1/2}\mathbf{V}^T\ \mathbf{V\Lambda V}^T\ \mathbf{V\Lambda}^{-1/2} = \mathbf{\Lambda}^{-1/2}\mathbf{\Lambda\Lambda}^{-1/2} = \mathbf{I} \quad ,$$

and (81) reduces to an identity matrix

$$\overline{\mathbf{D}} = \mathbf{I} \,. \tag{82}$$

Substituting (82) into (80) gives the final form for the maximum likelihood model covariance matrix:

$$\overline{\mathbf{C}} = \overline{\mathbf{H}}\,\overline{\mathbf{H}}^T \quad . \tag{83}$$

It should be noted that for the special case (64) of a diagonal data covariance matrix with equal variances, (83) is equal to the least squares model covariance matrix (65).

In terms of the filtered inverses (49-51), the maximum likelihood model covariance can be found by substituting into (83). Assume that the original problem has been transformed to a maximum likelihood problem by (76), defining the new least squares problem

$$\overline{\mathbf{y}} = \overline{\mathbf{A}}\mathbf{x} + \overline{\boldsymbol{\epsilon}} \ .$$

Let

$$\overline{\mathbf{A}}\mathbf{F}^{-1} = \hat{\mathbf{A}} = \hat{\mathbf{U}}\hat{\boldsymbol{\Lambda}}\hat{\mathbf{V}}^T \quad .$$

The covariance matrices for (49-51) are

$$\overline{\mathbf{C}}_{fd} = \mathbf{F}^{-1}\,(\hat{\mathbf{A}}^T\hat{\mathbf{A}} + \gamma^2\mathbf{I})^{-1}\,\hat{\mathbf{A}}^T\hat{\mathbf{A}}\,(\hat{\mathbf{A}}^T\hat{\mathbf{A}} + \gamma^2\mathbf{I})^{-1}(\mathbf{F}^{-1})^T \tag{84}$$

$$\overline{\mathbf{C}}_{fc} = \mathbf{F}^{-1}\,\hat{\mathbf{A}}^T(\hat{\mathbf{A}}\hat{\mathbf{A}}^T + \gamma^2\mathbf{I})^{-1}(\hat{\mathbf{A}}\hat{\mathbf{A}}^T + \gamma^2\mathbf{I})^{-1}\,\hat{\mathbf{A}}\,(\mathbf{F}^{-1})^T \tag{85}$$

$$\overline{\mathbf{C}}_{fs} = \mathbf{F}^{-1}\,\hat{\mathbf{V}}\,\check{\boldsymbol{\Lambda}}^{-2}\,\hat{\mathbf{V}}^T\,(\mathbf{F}^{-1})^T \tag{86}$$

where

$\check{\boldsymbol{\Lambda}}^{-2}$ = $n\,\mathrm{x}\,n$ diagonal matrix with $k$ non-zero elements defined by

$$\check{\boldsymbol{\Lambda}}_j^{-2} = \hat{\lambda}_j^2/(\hat{\lambda}_j^2 + \gamma^2)^2 \quad .$$

For non-zero $\gamma$, all these forms are equivalent. If $\mathbf{F}$ is an identity matrix, (84-86) reduce to equivalent stochastic covariance matrices.

### Model resolution

The Backus-Gilbert method, outlined in equations (3-8), can also be applied to discrete matrix problems (Menke, 1984, p. 61; Twomey, 1977, p. 169). Assume the problem can be written as

$$\mathbf{y} = \mathbf{A}\,\mathbf{x} \; .$$

Multiply both sides by **H**, an unknown inverse matrix

$$\mathbf{H}\,\mathbf{y} = \mathbf{H}\,\mathbf{A}\,\mathbf{x} \; .$$

This can be written as

$$\mathbf{H}\,\mathbf{y} = \mathbf{R}\,\mathbf{x} \; , \tag{87}$$

where

$$\mathbf{R} = \mathbf{H}\,\mathbf{A} \tag{88}$$

is the resolving kernel of the system. If the elements of the **H** matrix can be found such that

$$\mathbf{R}\,\mathbf{x} \approx \mathbf{I}\,\mathbf{x} = \hat{\mathbf{x}} \; , \tag{89}$$

then substituting into (87) gives the solution

$$\hat{\mathbf{x}} = \mathbf{H}\,\mathbf{y} \; .$$

From (89), the resolving kernel constructs the estimate $\hat{\mathbf{x}}$ as an average over the solution space **x**.

The Backus-Gilbert method differs from least squares and maximum likelihood inversion in that the inverse is found independently of the data. The elements of **H** are constructed by minimizing the difference, subject to variance constraints, between the resolving kernel and an identity matrix, instead of between observed and theoretical data. However, maximum likelihood and least squares inverses can be used to define the resolving kernel (88). The meaning is the same as in (89), except that the concept of resolution is not directly addressed in constructing the inverse matrices.

Assume that the original problem has been transformed to a maximum likelihood problem by (76), defining a new least squares problem

$$\overline{\mathbf{y}} = \overline{\mathbf{A}}\,\mathbf{x} + \overline{\boldsymbol{\epsilon}} \ .$$

From (88), the resolving kernel of the transformed system is

$$\overline{\mathbf{R}} = \overline{\mathbf{H}}\,\overline{\mathbf{A}} \ . \tag{89}$$

For filtered inverses, let

$$\overline{\mathbf{A}}\,\mathbf{F}^{-1} = \hat{\mathbf{A}} = \hat{\mathbf{U}}\hat{\boldsymbol{\Lambda}}\hat{\mathbf{V}}^{T} \ .$$

Substituting (49-51) into (89) gives the resolving kernels:

$$\overline{\mathbf{R}}_{fd} = \mathbf{F}^{-1}\,(\hat{\mathbf{A}}^{T}\hat{\mathbf{A}} + \gamma^{2}\mathbf{I})^{-1}\,\hat{\mathbf{A}}^{T}\hat{\mathbf{A}}\,\mathbf{F} \tag{90}$$

$$\overline{\mathbf{R}}_{fc} = \mathbf{F}^{-1}\,\hat{\mathbf{A}}^{T}(\hat{\mathbf{A}}\hat{\mathbf{A}}^{T} + \gamma^{2}\mathbf{I})^{-1}\,\hat{\mathbf{A}}\,\mathbf{F} \tag{91}$$

$$\overline{\mathbf{R}}_{fs} = \mathbf{F}^{-1}\,\hat{\mathbf{V}}\,\hat{\boldsymbol{\Sigma}}\,\hat{\mathbf{V}}^{T}\,\mathbf{F} \tag{92}$$

where

$\hat{\Sigma}$ = $n$x$n$ diagonal matrix with $k$ non-zero elements defined by

$$\hat{\Sigma}_j = \hat{\lambda}_j^2/(\hat{\lambda}_j^2+\gamma^2) \ .$$

As in the case of the covariance matrices, (90-92) are equivalent for non-zero $\gamma$.

# References

Aki, K. and P. Richards (1980). *Quantitative Seismology: Theory and Methods,* W. H. Freeman and Co.

Backus, G. and F. Gilbert (1967). Numerical Applications of a Formulism for Geophysical Inverse Problems, *Geophys. J. R. astr. Soc. 17,* 247-276.

Backus, G. and F. Gilbert (1968). The resolving power of gross earth data, *Geophys. J. 16,* 169-205.

Jackson, D. D. (1972). Interpretation of inaccurate, insufficient and inconsistent data, *Geophys. J. 28,* 97-109.

Jenkins, Gwilym M., and D. G. Watts (1968). *Spectral Analysis and Its Applications,* Holden Day, San Francisco.

Jordan, T. H. and J. Franklin (1971). Optimal solutions to a linear inverse problem in geophysics, *Proceedings of the National Academy of Sciences 70,* 291-293.

Lawson, C. L., and R. J. Hanson (1974). *Solving Least Squares Problems,* Prentice-Hall, Inc., New Jersey.

Levenberg, L. (1944). A method for the solution of certain non-linear problems in least squares, *Quart. Appl. Math. 2,* 164-168.

Marquardt, D. W. (1963). An algorithm for least-squares estimation of nonlinear parameters, *J. Soc. Indust. Appl. Math. 11, No. 2,* 431-441.

McMechan, G. A. (1983). Sesmic tomography in boreholes, *Geophys. J. R. Astr. Soc. 74,* 601-612

Menke, W. (1984). *Geophysical Data Analysis: Discrete Inverse Theory,* Academic Press, Florida.

Twomey, S. (1977). *Introduction to the Mathematics of Inversion in Remote Sensing and Indirect Measurements,* Elsevier Scientific Publishing Co., Amsterdam.

Wiggins, R. (1972). The general linear inverse problem: implications of surface waves and free oscillations on earth structure, *Rev. Space Phys. 10,* 251-284.

Woodhouse, J. H. and A. M. Dziewonski (1984). Mapping the upper mantle: three dimensional modeling of the earth structure by inversion of seismic waveforms, *J. Geophys. Res., 89, B7,* 5953-5986.

# APPENDIX III

## 30.0 FREQUENCY VARIABLE FILTERS.

30.1 Purpose. This appendix has been included to provide the theory of the frequency-variable filter (renamed from the time-variable filter) incorporated in the Phase-Match Filter function of the SWANS.

30.2 Origin. These pages are a pre-print of a paper submitted to the *Bulletin of the Seismological Society of America*.

# APPLICATION OF FREQUENCY VARIABLE FILTERS TO SURFACE-WAVE AMPLITUDE ANALYSIS

By David R. Russell, Robert B. Herrmann, and Horng-Jye Hwang

# ABSTRACT

The problem of spectral biasing due to frequency domain filtering of surface-wave seismograms is investigated, and the method of frequency variable filters (FVF) is developed to compensate for this bias. A detailed comparison of some currently accepted surface-wave filters is made, in order to clarify the development of the FVF algorithm. Three long period surface-wave seismograms are tested with FVF and compared to two other methods, the multiple filter technique and phase matched filters. Emphasis is placed on finding limitations in all the methods, not on routine processing.

## Introduction

Recently, attention has been focused on the problem of filtering seismograms to isolate propagating normal modes, especially in the context of determining spectral amplitudes for nuclear yield estimation. Yacoub (1983) utilized the multiple filter technique (MFT) to determine average spectral amplitudes around the 20 second period. Herrin and Goforth (1977) developed the phase-matched filter (PMF) to refine group and phase velocities of normal modes, and Stevens (1986) used this method to isolate fundamental mode spectra across the entire observed bandwidth. Hwang and Mitchell (1986) used the time-variable filter (TVF) developed by Landisman *et al.* (1969) to isolate spectral amplitudes for inter-station Green's functions. Russell *et al.* (1986) combined the PMF and TVF methods to form a frequency-variable filter (FVF), in order to improve spectral amplitude estimates of explosion generated waveforms.

This paper will extend the FVF method in order to reduce problems of spectral biasing, and will discuss in some detail relationships between the above filters.

## Theory

For surface-wave studies, the objective is to process $s(t)$, the raw surface-wave time history, by a series of convolutional filters designed for modal isolation. In the context of this discussion, the definition of "convolution" is extended to include the general case of frequency varying filters, that is, filters that have a changing bandwidth during the convolution operation. This will produce a filtered trace (or spectrum) which can be used for further analysis. Most surface-wave filters can be expressed by the following relations:

$$\psi(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} P(\omega)\, W_r(t,\omega)\, S(\omega)\, e^{i\omega t}\, d\omega \tag{1}$$

$$\Psi(\omega) = \int_{-\infty}^{\infty} \psi(t)\, e^{-i\omega t}\, dt \tag{2}$$

$$h(t,\omega_0) = \frac{1}{2\pi} \int_{0}^{\infty} \Psi(\omega)\, H(\omega - \omega_0)\, e^{i\omega t}\, d\omega \tag{3}$$

where

$$S(\omega) = \sum_j |S_j(\omega)|\, e^{-ik_j(\omega)x}$$

$$P(\omega) = e^{i\tilde{k}_j(\omega)x}$$

$W_r(t,\omega) =$ time and frequency variable window

$H(\omega - \omega_0) =$ frequency domain convolution filter.

$S(\omega)$ is the total spectrum of the seismogram, $s(t)$, and is composed of a sum of normal modes, multi-pathed signals, possible interfering events and phases, and incoherent noise. The purpose of the above filters is to isolate the j'th mode of interest so the amplitude spectrum $|S_j(\omega)|$ and wavenumber spectrum $k_j(\omega)$ can be recovered. This is probably too idealistic a goal in some cases, as pointed out by Der (1986). D,ue to scattering and reflections, there may not be a pure isolated mode to recover. A more accurate statement is that the purpose of such filters is to isolate seismic energy propagating in the vicinity of desired modes of interest.

$P(\omega)$ is a phase-matched filter. The wavenumber estimate $\tilde{k}_j$ should be near the true modal wavenumber, in order to compress the energy of the desired signal about zero-lag in the time domain, forming a "pseudo-autocorrelation function" $\psi(t)$. Herrin and Goforth (1977) discussed this in detail.

$W_\tau(t,\omega)$ is a time and frequency variable window used to isolate modes of interest and improve signal to noise ratios. It is symmetric about position $\tau$ in the time domain, with a width controlled by the frequency $\omega$.

Various combinations of the above filters have been used for modal isolation, five of which will be detailed below.

Case 1: $P(\omega) \equiv 1$ , $W_\tau(t,\omega) \equiv 1$

This is the basis for the multiple filter technique (MFT) (Dziewonski *et al.*, 1969). Equations (1) and (2) are simply Fourier transforms, so the raw spectrum $S(\omega)$ is input into (3), which is the MFT evaluated at $\omega_0$. $H(\omega)$ is a narrow bandpass filter (usually a band-limited Gaussian), symmetrically about $\omega_0$. The non-negative integral limits cause the time signal $h(t,\omega_0)$ to be complex, with the modulus having maxima at the group velocities of the signal modes. Herrmann (1973) showed that under the condition of an approximately flat amplitude spectrum and linear phase delay of the j'th mode across the width of $H(\omega)$,

$$h(t_j,\omega_0) \approx A \; |S_j(\omega_0)| \; e^{i[\omega_0 t_j - k_j(\omega_0)x]} \tag{4}$$

where $A$ is a constant of proportionality determined by the frequency and the width of the Gaussian filter $H(\omega)$, and $t_j$ is the group delay of the j'th mode. Evaluating (3) at multiple frequencies will extract the spectrum of the j'th mode, if it is suitably smooth.

Case 2: $W_r(t,\omega) \equiv 1$

This is the basis for the "residual dispersion" technique of Dziewonski *et al.* (1972). The MFT is first applied to determine wavenumber estimates from the instantaneous phases of (4), in order to construct the phase-matched filter $P(\omega)$. Equation (1) now represents the seismogram with the dispersion removed from the j'th mode. The signal is Fourier transformed (2) and the MFT applied again (3). This method was introduced to remove biasing caused by the signal phase in MFT due to dispersion.

Case 3: $P(\omega) \equiv 1, \quad W_r(t,\omega) = W_{t_j(\omega)}(t,\omega)$

This is the time-variable filter due to Landisman *et al.* (1969). MFT is performed first to determine group delays $t_j(\omega)$ , and the window $W_r$ is symmetrically centered about the group delay in equation (1). The width of the window is a multiple of the period of interest. Thus $\psi(t)$ should contain only energy associated with the mode of interest. Hwang and Mitchell (1986) used equation (2) to represent isolated normal mode spectra

$$\Psi(\omega) \approx S_j(\omega) \ .$$

Case 4: $W_r(t,\omega) = W_0(t)$

This is the phase-matched filter method of Herrin and Goforth (1977). MFT is performed first to estimate wavenumbers for the phase-matched filter $P(\omega)$. This differs from the residual dispersion technique (case 2), in that the phase of the signal is found by integrating the group delay calculated in MFT. The window $W_0(t)$ is no longer frequency dependent, so it can be factored from the integral in (1). It is centered about zero-lag in the time domain. Equation (1) is now the windowed pseudo-autocorrelation function, and (2) is the phase-matched spectrum of the isolated mode of interest.

All of the above methods attempt to isolate spectral modes of interest via frequency domain convolutions. As Dziewonski and Hales (1972) pointed out, the process of convolution will distort the signal spectrum unless the prescribed filters are Dirac impulses in the frequency domain. The process of residual dispersion (case 2), or phase-matched filtering (case 4) can minimize phase distortion, but they do not address the problem of amplitude distortion, or bias due to the convolution operation.

The method presented in this paper (FVF) will combine the beneficial aspects of a time-variable filter (TVF) and a phase-matched filter (PMF), and will allow the analyst to place maximum error bounds on the amount of tolerable amplitude bias.

Case 5: $W_r(t,\omega) = W_0(t,\omega)$

This is the method of frequency variable filtering (FVF). The raw spectrum $S(\omega)$ is phase-matched filtered to compress the energy of the signal of interest about zero-lag in the time domain. Then, each harmonic component of this pseudo-autocorrelation function is windowed about zero-lag with $W_0(t,\omega)$, with the width of the window proportional to the period of the individual harmonic. These operations are done in equation (1), which is then Fourier transformed by equation (2) to give the isolated modal spectrum of interest.

This technique is quite similar to the time-variable filter (case 3), except that the phase of the mode of interest is first removed to minimize spectral distortions due to phase fluctuations. It should be noted that in the context of residual dispersion measurements, this method was first recommended by Dziewonski and Hales (1972).

An advantage of FVF is that the time window $W_0(t,\omega)$ can be constructed at each frequency to control the amount of bias due to windowing. The exact form of the window will be defined below, but first a discussion of the problem of biasing is presented.

### Window bias

In calculating the Fourier transform (2), it is assumed that the window $W_r(t,\omega)$ does not distort the spectrum of the j'th mode. This is not true in practice, and the purpose of this section is to approximately calculate the bias in the frequency domain due to time domain windowing. The analysis is similar to Jenkins and Watts (1968, p. 247). From equation (1), define the signal pseudo-autocorrelation function as a phase-matched single-mode signal uncontaminated by noise:

$$\psi_j(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} |S_j(\omega)|\, e^{i\delta k(\omega)\, z}\, e^{-i\omega t}\, d\omega \tag{5}$$

where

$$\delta k(\omega) \;=\; \tilde{k}_j(\omega) - k_j(\omega)$$

is the residual wavenumber left from the phase-matched filter process. The bias in the frequency domain will be the transformed difference between the windowed signal pseudo-autocorrelation function and (5):

$$B(\omega) = \int_{-\infty}^{\infty} [W_0(t) - 1]\, \psi_j(t)\, e^{-i\omega t}\, dt. \tag{6}$$

At this point it is necessary to define the particular window desired. Harris (1978) reviewed a broad class of spectral windows and their characteristics. One measure of window performance is the relative sidelobe level. This gives an indication of the "smoothness" or convolutional rippling effect in the frequency domain. When there is significant truncation of signal (or noise) in the time domain by the window, the transformed window has high sidelobes, which causes distortional rippling in the frequency domain. The rectangular window is the worst, with a maximum sidelobe -13 dB below (almost 1/2) the main lobe. However, Jenkins and Watts (1968) showed that windows that tend toward the rectangular have the least signal bias, under the condition that there is no truncation of signal or noise in the time domain.

Two windows are examined for bias: the cosine and Parzen (also called de la Valle - Poisson) windows. The cosine window is defined as:

$$W_o^c(t) = \begin{cases} \cos(\pi t/2T) & |t| \le T \\ 0 & |t| > T \end{cases} \tag{7}$$

where $T$ is the one-sided width of the cosine window. The maximum sidelobe level for the cosine window is -23 dB (7 percent) of the main lobe. The Parzen window is defined as:

$$W_o^p(t) = \begin{cases} 1 - 6(t/T)^2 + 6(|t|/T)^3 & |t| \leq T/2 \\ 2(1 - |t|/T)^3 & T/2 < |t| \leq T \\ 0 & |t| > T \end{cases} \tag{8}$$

where $T$ is the one-sided width of the Parzen window. The maximum sidelobe level for the Parzen window is -53 dB (0.2 percent) of the main lobe. For a given width $T$, the Parzen window is a much smoother convolutional filter in the frequency domain due to low sidelobes.

To calculate the approximate bias, substitute the windows into (6) and keep terms only on the order of $1/T^2$ or more. For the cosine window,

$$B^c(\omega) = \frac{\pi^2}{8T^2} \int_{-\infty}^{\infty} -t^2 \, \psi_j(t) \, e^{-i\omega t} \, dt \quad + \; O(1/T^4)$$

and the Parzen window,

$$B^p(\omega) = \frac{6}{T^2} \int_{-\infty}^{\infty} -t^2 \, \psi_j(t) \, e^{-i\omega t} \, dt \quad + \; O(1/T^3) \, .$$

Notice that the infinite limits are kept for the integrals. This is under the assumption that $T$ is wide enough to insure an insignificant truncation of the signal pseudo-autocorrelation function, resulting in a negligible signal outside the limits $+T$, $-T$ . Making use of Fourier transform properties of differentiation (Papoulis, 1962, p. 16),

$$B^c(\omega) = \frac{\pi^2}{8T^2} \, \Psi_j''(\omega) \tag{9}$$

$$B^{p}(\omega) = \frac{6}{T^2}\,\Psi_j''(\omega) \quad , \tag{10}$$

where the double primes indicate the second derivative with respect to angular frequency of the signal pseudo-autocorrelation spectrum. Taking the ratio of (9) to (10) shows that the Parzen window has almost five times the bias of the cosine window, which suggests that it may be a poor choice for a windowing function. However, further investigation of the spectral second derivative yields interesting results.

From equation (5), the spectrum of the signal pseudo-autocorrelation function is

$$\Psi_j(\omega) \;=\; |S_j(\omega)|\, e^{i\delta k(\omega) z} \;=\; \alpha e^{i\theta} . \tag{11}$$

The terms $\alpha$ and $\theta$ are chosen for notational convenience. Calculating the second derivative of the signal pseudo-autocorrelation spectrum gives

$$\Psi_j''(\omega) = [\alpha'' - \alpha(\theta')^2]e^{i\theta} + [\alpha\theta'' + 2\alpha'\theta']e^{i(\theta + \pi/2)} . \tag{12}$$

If the phase-matched filter is successful, the difference between the estimate and true wavenumber ( $\delta k$ ) should approach zero. Therefore, keeping only first order residual phase terms for $\theta$, $\theta'$, and $\theta''$, the Parzen window amplitude bias is

$$B^{p\alpha} = \frac{6}{T^2}\,\alpha'' \tag{13}$$

and the phase bias is

$$B^{p\theta} = \frac{6(\alpha\theta'' + 2\alpha'\theta')}{T^2\alpha + 6\alpha''} . \tag{14}$$

A similar expressions for the cosine window amplitude bias is

$$B^{c\alpha} = \frac{\pi^2}{8T^2}\,\alpha'' \tag{15}$$

and for phase bias,

$$B^{c\theta} = \frac{\pi^2(\alpha\theta'' + 2\alpha'\theta')}{8T^2\alpha + \pi^2\alpha''} \quad . \tag{16}$$

Equation (14) and (16) show that the bias in phase is independent of constant phase values, and that if the first and second derivatives are small, the phase bias will also be small. Therefore, it is advisable to use smooth (low sidelobe) windows for calculating residual phases in the matched filtering process. However, it should be clear that spectral amplitudes may be quite biased when there is significant curvature in the amplitude spectrum, corresponding to a large spectral second derivative, $\alpha''$. Examples of this are narrow band spectra and the vicinity of sharply changing band edges. The fact that phase-matched filtering is an iterative process can reduce residual bias in phase, but this will not help the bias in the amplitude spectrum, since it is to first order independent of phase.

## FVF algorithm

To implement the FVF algorithm, first note that the $\Psi_j''$ term in equation (9) corresponds to the true signal pseudo-autocorrelation function, which is unknown. However, with the same analysis used to calculate (9), it can be directly shown that

$$B^c(\omega) = \frac{\pi^2}{8T^2}\Psi_j^{p\,''}(\omega) \; + O(1/T^4) \tag{17}$$

where the superscript " $p$ " indicates the second derivative of the Parzen windowed pseudo-autocorrelation function defined by (1). Define the maximum tolerable bias error relative to the maximum signal amplitude as

$$E_{\max} = \frac{|B|_{\max}}{|\Psi_j^p|_{\max}} \quad . \tag{18}$$

For the cosine window, substitute (18) into (17) and solve for $T^c$ , the one-sided width of the window:

$$T^c = \left[ \frac{\pi^2}{8E_{\max}} \frac{|\Psi_j^{p''}(\omega)|}{|\Psi_j^p|_{\max}} \right]^{\frac{1}{2}} \tag{19}$$

To account for zero-crossings of the second derivative, let

$$T^c(\omega) = \max \left( \frac{2\pi C}{\omega} , T^c \right) \tag{20}$$

where $C$ is a constant multiple of the period of interest. Using equations (7) and (20), the instantaneous cosine window for the FVF algorithm is defined as

$$W_o^c(t,\omega) = \begin{cases} \cos\left[\pi t/2T^c(\omega)\right] & |t| \leq T^c(\omega) \\ 0 & |t| > T^c(\omega) \;. \end{cases} \tag{21}$$

The FVF algorithm can now be formally developed as follows.

**Step 1.** Using Herrin and Goforth's iterative phase-matched filter process (1977), isolate the mode of interest with a Parzen windowed pseudo-autocorrelation function

$$\psi_j^p(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} P_j(\omega)\; W_o^p(t)\; S(\omega)\; e^{i\omega t}\, d\omega \quad . \tag{22}$$

and save the final wavenumber estimate used in the phase-matched filter

$$P_j(\omega) = e^{i\bar{k}_j(\omega)x} \quad . \tag{23}$$

**Step 2.** From equation (22), calculate the spectrum of the windowed mode

$$\Psi_j^p(\omega) = \int_{-\infty}^{\infty} \psi_j^p(t)\, e^{-i\omega t}\, dt \tag{24}$$

and its second derivative

$$\Psi_j^{p\,''}(\omega) = \int_{-\infty}^{\infty} -t^2\, \psi_j^p(t)\, e^{-i\omega t}\, dt \quad . \tag{25}$$

Substitute equations (24) and (25) into (19) for the instantaneous window width $T^c(\omega)$.

**Step 3.** Calculate the corrected pseudo-autocorrelation function and its spectrum, using the phase-matched filter (23) and the instantaneous cosine window (21):

$$\psi_j(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} P_j(\omega)\; W_o^c(t,\omega)\; S(\omega)\; e^{i\omega t}\, d\omega \tag{26}$$

$$\Psi_j(\omega) = \int_{-\infty}^{\infty} \psi_j(t)\, e^{-i\omega t}\, d\omega \quad . \tag{27}$$

**Step 4.** Using the phase-matched filter (23) and the corrected pseudo-autocorrelation spectrum (27), calculate the corrected spectrum of the isolated mode:

$$S_j(\omega) = \Psi_j(\omega)\, e^{-i\tilde{k}_j(\omega)\, x} \quad . \tag{28}$$

This completes the FVF algorithm.

## Examples and discussion

Three seismic events were chosen to illustrate various aspects of the

FVF algorithm and its relation to other convolutional type filters. For comparison, MFT and PMF were also applied to the events. The MFT (3) was constructed with a Gaussian filter

$$H(\omega) = \begin{cases} \exp(-\alpha\, \omega^{2}/\omega_0^{2}) & |\omega| \leq \omega_c \\ 0 & |\omega| > \omega_c \end{cases} .$$

The Gaussian cutoff frequency $\omega_c$ and the filter width parameter $\alpha$ were constructed with values found in Dziewonski *et al.* (1969) and Herrmann (1973),

$$\omega_c = \omega_0/4 \qquad \alpha = 16\pi \; .$$

The residual dispersion method (Case 2) was not incorporated into the MFT, in order to illustrate the effects of phase distortion.

The PMF was implemented using the algorithm developed by Herrin and Goforth (1977). The Parzen window (8) was used for phase processing, and the cosine window (7) was used to extract the amplitude spectrum. The one-sided window width was set to a default value of 1.5 times the maximum period found by MFT analysis.

The FVF was constructed using the algorithm developed above. Parameters $E_{\max}$ and $C$ in (18) and (20) were set to

$$E_{\max} = .035 \qquad C = 2.5 \; .$$

This defines a minimum one-sided cosine window width of 2.5 times the period of interest, and a maximum bias error of 3.5% . As stated above, some error must be tolerated in the filtering process. Setting $E_{\max} = 0$ would simply turn the FVF into an allpass filter. The problem of bias error is also inherent in both PMF and MFT filters, as will be shown.

The advantage of FVF is that the bias can be controlled.

To realize the PMF and FVF filters, a computer program was written with interactive graphics capability. The width of the time windows used in the filters can be controlled by visual inspection of the pseudo-autocorrelation functions. Inputs to the program are the digitized seismogram and group velocity estimates from MFT for the mode of interest. The outputs are corrected group and phase velocities, and the complex spectrum of the isolated mode.

The sample seismograms were chosen to illustrate and contrast the above filters, and should not be considered as "typical" events for processing. The seismograms were generated by earthquake sources, and no attempt was made to remove the spectral effects of the source mechanism or the source time function from the events. However, instrument deconvolution was performed.

Event I is a synthetic seismogram recorded at a distance of 1000 kilometers from a 45 degree pure dip-slip source mechanism at a depth of 50 kilometers. The seismogram is composed of vertical component fundamental and first higher mode Rayleigh waves. Figure 1 (top) is the amplitude spectrum of the signal, showing clear spectral contamination of the fundamental mode by the first higher mode at periods less than 20 seconds. The bottom figure is the multiple filter contour map (Dziewonski *et al.* , 1969) giving the distribution of seismic energy as a function of period and group velocity.

For MFT processing, maximum values on the contour map were picked for the fundamental mode, and spectral amplitudes were calculated using equation (4). Group velocities were picked from the maximum amplitude points on the contour map and used as input for PMF and FVF processing.

Figure 2 shows the final pseudo-autocorrelation functions for PMF and FVF processing. A cosine window with a one-sided width of 20 seconds was required to exclude the higher mode in the PMF, and this width was also used for processing bias estimates in FVF. Notice that the width of the FVF pseudo-autocorrelation function appears to be slightly wider than the PMF in Figure 2. Figure 14 shows the actual window width as a function of period. The FVF window changes as a multiple of the period of interest (equation 20), and in this case, no curvature correction was necessary.

Figures 3 and 4 show the time domain results of PMF and FVF, respectively. The isolated fundamental mode for each process appears almost identical. However, the residual higher mode for the PMF (Figure 3) shows some fundamental mode contamination. Figure 5 is more informative, contrasting the difference between the theoretical fundamental mode and the filtered amplitude spectra for the 3 processes. The MFT and FVF are almost identical to the theoretical spectrum, while PMF exhibits biasing over the entire spectrum. It is most pronounced at 30 seconds, which corresponds to the region of highest curvature.

Event I is somewhat extreme, in that the presence of the higher mode forces a narrow bandwidth for the PMF window, causing a large bias as predicted by equation (9). In practice, when dealing with smoothly varying amplitude spectra due to shallow depth explosions, higher modes may not be significantly excited, and the PMF can be constructed with time windows exceeding 100 seconds. This was graphically demonstrated in

Stevens' reply to Der (1986, p. 1828). Long period spectral curvature was slight for the event Stevens chose, and there was no apparent long period bias for two-sided window widths varying from 50 to 1000 seconds. This indicates that in that case, PMF was justified for spectral amplitude determination.

To demonstrate the effect of sharply varying curvature on the filters, events II and III, with pronounced spectral nulls due to double-couple source mechanisms, were picked. Both seismograms are actual events recorded on long period WWSSN vertical component seismometers. Event II was recorded at station SHI in Iran, from a source located in the Gulf of Aden, with the signal propagating across the southeastern Arabian Plate.

This seismogram is composed of two superimposed events, the initial earthquake and a stronger event occurring approximately 175 seconds after the original. The multiple filter contour map in Figure 6 separates the superimposed events into two distinct energy bands, which are almost identical in shape, indicating a similar source mechanism and propagation path.

Pseudo-autocorrelation functions for PMF and FVF analysis are given in Figure 7. The windowed results are almost identical, so only the FVF isolated mode is shown in Figure 8. This figure graphically demonstrates the power of matched filters to effectively separate superimposed signals, in addition to improving the signal to noise ratio.

Since there is no theoretical reference spectrum, the filtered amplitude spectra were simply superimposed on each other in Figure 9 for comparison. The most significant difference is in the MFT spectrum, at 9 seconds and between 30-40 seconds. At 9 seconds, the Gaussian filter (29) has approximate half-amplitude values ( $e^{-.7}$ ) at 8 and 10.2 seconds. From Figure 9, it is clear that the spectrum has approximately the same bandwidth as the Gaussian filter at this point, which violates the requirement for a flat amplitude spectrum across the passband in equation (4). This results in the observed low amplitude. Between 30 and 40 seconds, applying Dziewonski's residual dispersion method (case 2) causes the discrepancy in this period range to disappear, indicating that there is enough slope in the group velocity (see Figure 6) to introduce significant phase distortion into MFT amplitude analysis.

The effect of curvature corrections in the FVF can be seen in Figure 14. There is an increase in window width at 9, 12, and 17 seconds, corresponding to the spectral peaks and nulls of Figure 9. It should be noted that the FVF behaves as poorly as the MFT at 9 seconds, if curvature corrections are not incorporated.

Event III was recorded at U.S. west coast station COR, from an earthquake occurring in the Aleutian Islands. The distance between source and receiver is 4005 kilometers, and the propagation path is oceanic, as seen on the MFT group velocity contour plot (Figure 10). The spectral plot in Figure 10 exhibits two distinct nulls in the signal passband, at 18 and 25 seconds. The null at 18 seconds results in a strong, narrowband spectral peak at 17 seconds, shown by the arrow on the spectral plot in Figure 10. This occurs in the period range where the group velocity slope is almost vertical, resulting in a strongly dispersed, sinusoidal time domain signal. This can be seen on the MFT contour map by comparing the energy of the signal at 17 seconds with the corresponding time domain signal on the right.

The event was picked to illustrate how all of the above filtering methods can fail, if the entire passband is considered to be the desired signal. This can be seen from the PMF and FVF pseudo-autocorrelation functions in Figure 11. The windowed functions delete a considerable portion of the signal energy, seen to the left of zero-lag on the raw pseudo-autocorrelation function. This energy corresponds to the 17 second peak found in the original spectrum. The reason for the misalignment is due to the initial estimate of group velocities from the MFT. At the 18 second null, there is a zero crossing in the complex spectrum, resulting in an instantaneous phase change of $\pi$ radians. The group delay at this point should be impulsive, since it is the derivative of the phase with respect to angular frequency (Papoulis, 1962, p. 134), and this should be seen as an impulse in group velocity. However, the Gaussian MFT filter smooths the impulse, causing a distortion of the group delay in the adjacent 17 second peak. As a result, the PMF is constructed incorrectly on the first iteration, with the 17 second energy to the left of zero-lag.

It is possible for the PMF to correct the signal phase, but as seen from Figure 11, this would require the time window to be at least twice as wide as shown, reducing considerably the ability of the filter to distinguish signal from noise. Figure 12 shows the effect of losing the 17 second peak on the isolated PMF and FVF modes, using the given windows.

Figure 13 shows the filtered spectra superimposed on the original spectrum, with the residuals between the two plotted below. Since all three methods lost the 17 second peak, the residual plots start at the 18 second null. The MFT shows the effect of phase distortion on the amplitude spectrum, with the large residual occurring at 20 seconds. This can be confirmed by observing the near vertical slope to the group velocity dispersion in Figure 10. This again points out the necessity for the residual dispersion method in MFT amplitude analysis.

The effect of curvature correction on the FVF cosine windows can be seen in Figure 14. Maximum widths at 18, 20, and 25 correspond to the peaks and nulls of the spectrum in Figure 10. It would seem that the largest width should correspond to the point of sharpest curvature in the spectrum, at 17 seconds. However, a review of equation (17) shows that the bias estimates are constructed from the Parzen windowed pseudo-autocorrelation function. Examination of Figure 11 again shows that the window would have to be widened considerably to extract this portion of the spectrum.

## Conclusion

The method of frequency variable filters (FVF) is a viable alternative to both the MFT and PMF techniques of extracting normal mode amplitudes from propagating multi-mode surface-waves. It has the advantage of explicitly addressing the problem of spectral bias, which is an unavoidable side effect of any type of convolutional smoothing in the frequency domain. FVF will not be successful on all types of surface-wave spectra, as indicated by event III. This particular case illustrates the fundamental tradeoff between frequency and time domain resolution. To successfully extract the 17 second peak, the time windows would have to be so wide that the filters would be essentially allpass. Both MFT and PMF had the same problem with this type of event.

FVF and PMF have an advantage over MFT, in that the results of processing can be seen both in the time and frequency domain. The pseudo-autocorrelation functions can be used as diagnostic tools, and the final isolated mode can be compared to the original seismogram. In addition, phase distortion can have a significant effect on the MFT, requiring application of a phase-matched filter in the form of the residual dispersion method (case 2). Both PMF and FVF remove the phase as an automatic part of processing. For the above reasons, it is recommended that the FVF, and in some cases the PMF, be preferred over the MFT as filters for isolating surface-wave normal mode amplitude spectra.

## Acknowledgements

The authors are grateful to Hafidh A. A. Ghalib and Hanan Al-Khatib for bringing the WWSSN seismograms used in this paper to our attention. This work was supported by the AFSC under Contract F19628-85-K-0029.

## References

Der, Z. A. (1986). Comments on the paper "Estimation of scalar moments from explosion-generated surface waves" by Jeffry L. Stevens, *Bull. Seism. Soc. Am. 76,* 1822-1824.

Dziewonski, A., S. Bloch, and M. Landisman (1969). A technique for the analysis of transient seismic signals, *Bull. Seism. Soc. Am. 59,* 427-444.

Dziewonski, A., and A. Hales (1972). Numerical analysis of dispersed seismic waves, *in* B. A. Bolt (editor), *Methods of computational physics, vol. 11,* Academic Press, New York, 39-85.

Dziewonski, A., J. Mills, and S. Bloch (1972). Residual dispersion measurement - a new method of surface-wave analysis, *Bull. Seism. Soc. Am. 62,* 129-139.

Harris, F. J. (1978). On the use of windows for harmonic analysis with the discrete Fourier transform, *Proceedings of the IEEE 66,* 51-83.

Herrin, E. and T. Goforth (1977). Phase-matched filters: application to the study of Rayleigh waves, *Bull. Seis. Soc. Am. 67,* 1259-1275.

Herrmann, R. B. (1973). Some aspects of band-pass filtering of surface waves, *Bull. Seism. Soc. Am. 63,* 663-671.

Hwang, H. J. and B. J. Mitchell (1986). Interstation surface wave analysis by frequency-domain Wiener deconvolution and modal isolation, *Bull. Seism. Soc. Am. 76,* 847-864.

Jenkins, Gwilym M., and D. G. Watts (1968). *Spectral Analysis and Its Applications,* Holden Day, San Francisco.

Landisman, M., A. Dziewonski, and Y. Sato (1969). Recent improvements in the analysis of surface wave observations, *Geophys. J. 17,* 369-403.

Papoulis, A. (1962). *The Fourier Integral and its Applications,* McGraw-Hill, New York.

Russell, D. R., H. J. Hwang, and R. B. Herrmann (1986). Matched filters vs. time variable filters, Abstract for the AFGL/DARPA Review of Nuclear Test Monitoring Research, U.S. Air Force Academy, Colorado Springs, Colorado, May 6-8 1986.

Stevens, J. L. (1986). Estimation of scalar moments from explosion-generated surface waves, *Bull. Seism. Soc. Am. 76,* 123-151.

Yacoub, N. K. (1983). "Instantaneous Amplitudes": a new method to measure seismic magnitude, *Bull. Seism. Soc. Am. 73,* 1345-1355.

Department of Earth and Atmospheric Sciences
Saint Louis University
P.O. Box 8099, Laclede Station
St. Louis, Missouri 63156

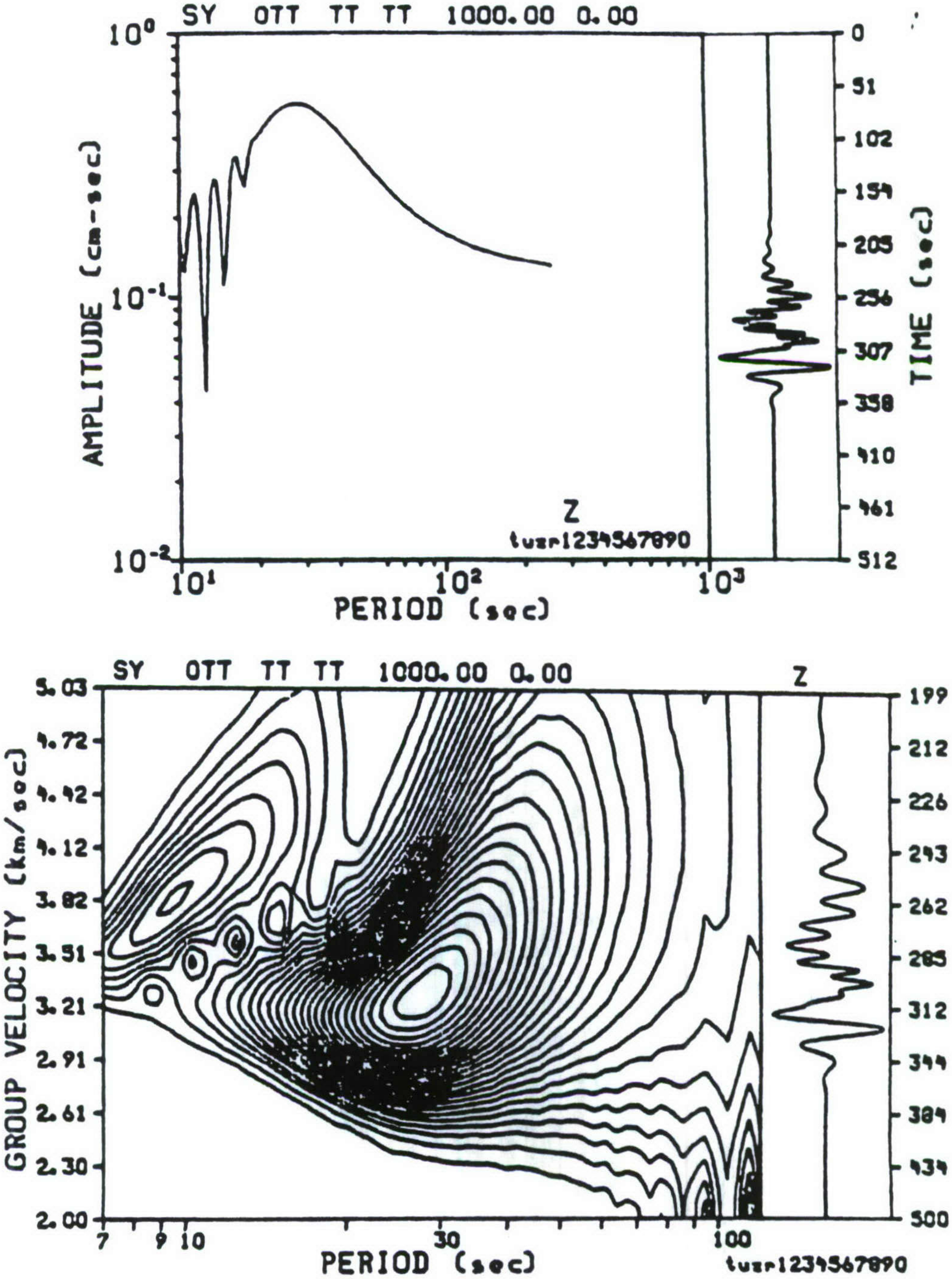

FIG. 1. Event I spectrum (top) and MFT group velocity contour plot (bottom). Time domain seismograms are plotted to the right. Notice the non-linear time scale for the MFT plot.

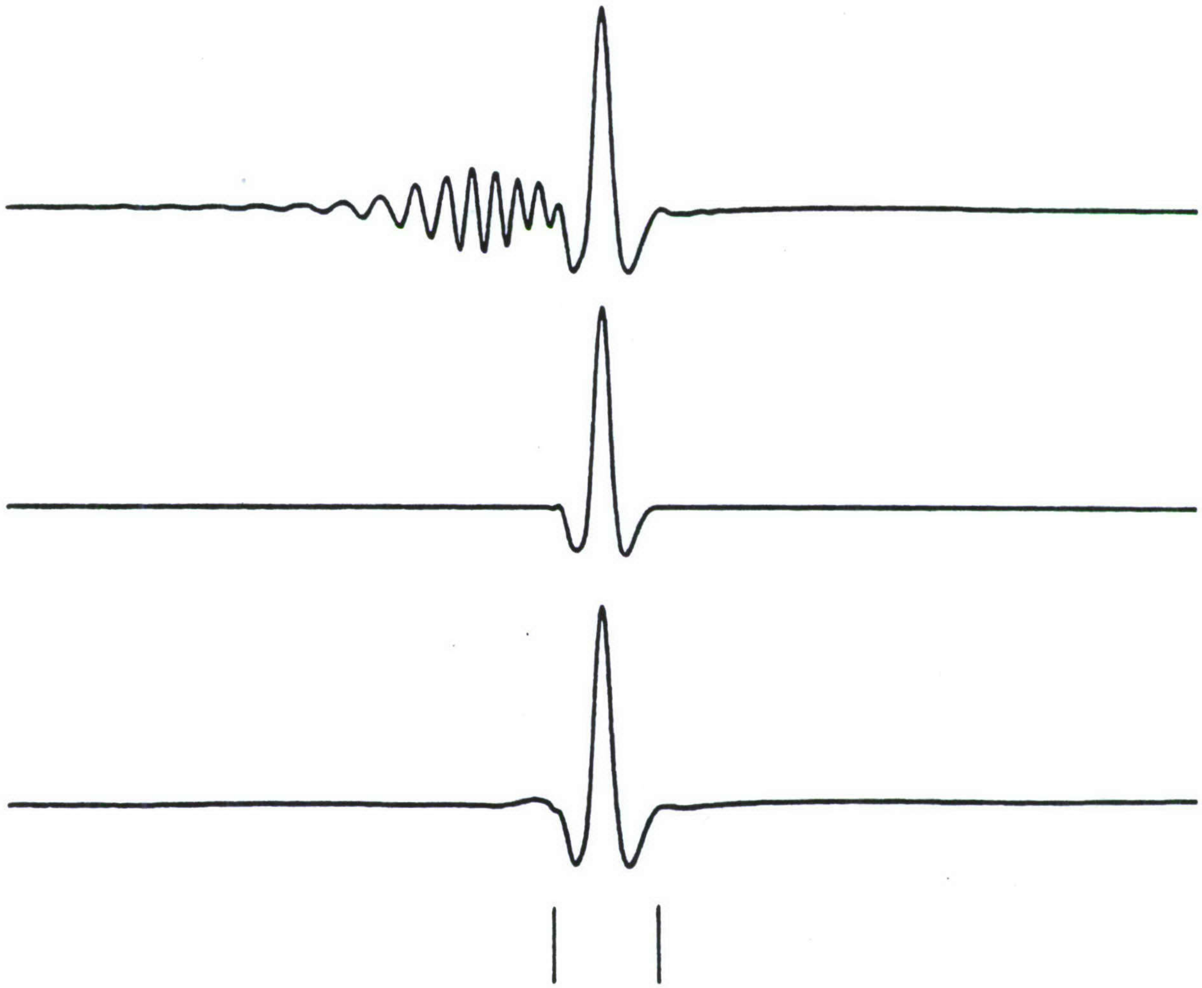

FIG. 2. Event I pseudo-autocorrelation functions. TOP: raw pseudo-autocorrelation function. MIDDLE: final pseudo-autocorrelation function for PMF. BOTTOM: final pseudo-autocorrelation function for FVF. The time marks indicate a two-sided window width of 40 seconds.

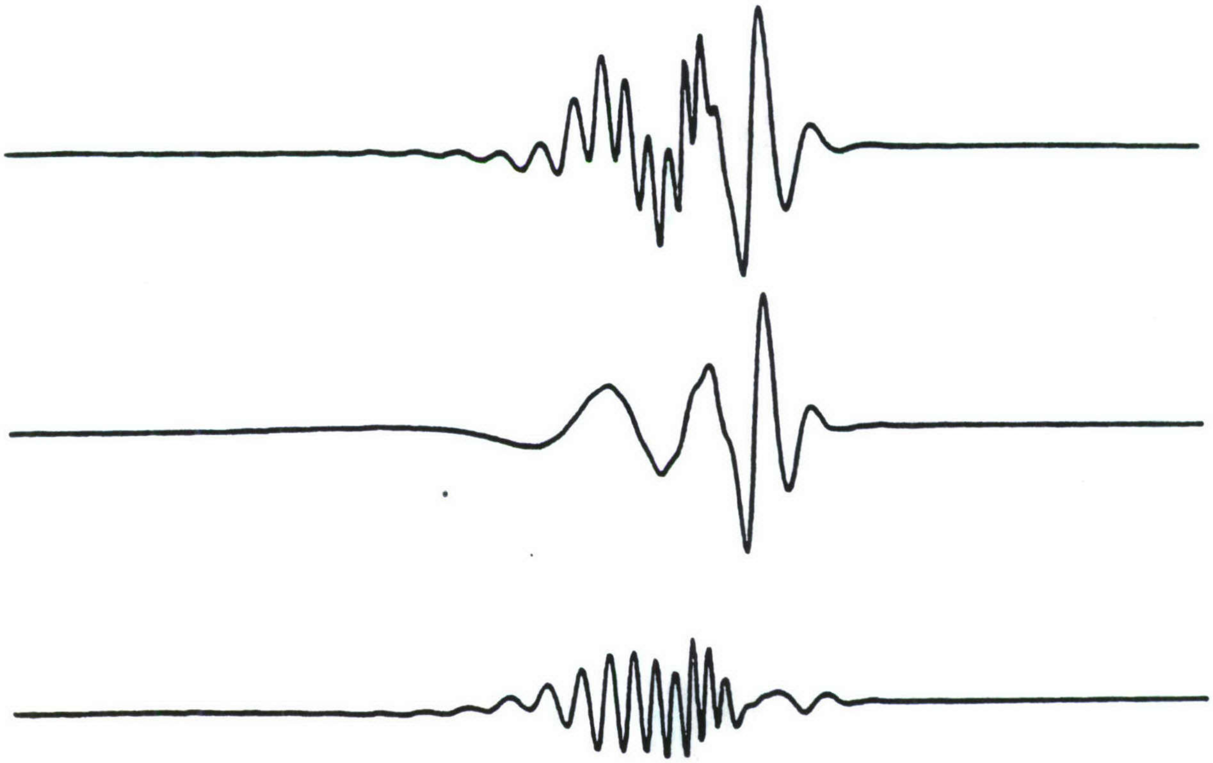

FIG. 3. Event I PMF seismograms. TOP: raw seismogram. MIDDLE: fundamental mode isolated by PMF. BOTTOM: residual obtained by subtracting the fundamental mode from the original seismogram.

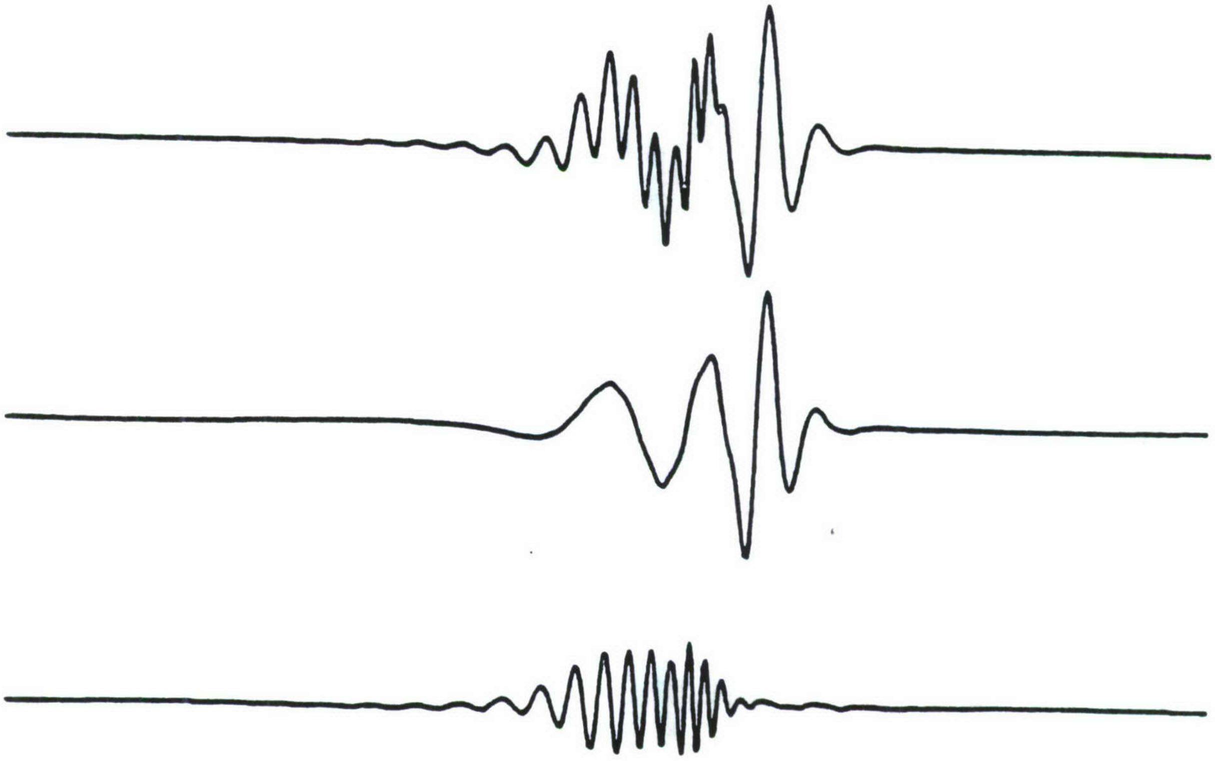

FIG. 4. Event I FVF seismograms. TOP: raw seismogram. MIDDLE: fundamental mode isolated by FVF. BOTTOM. residual obtained by subtracting the fundamental mode from the original seismogram.

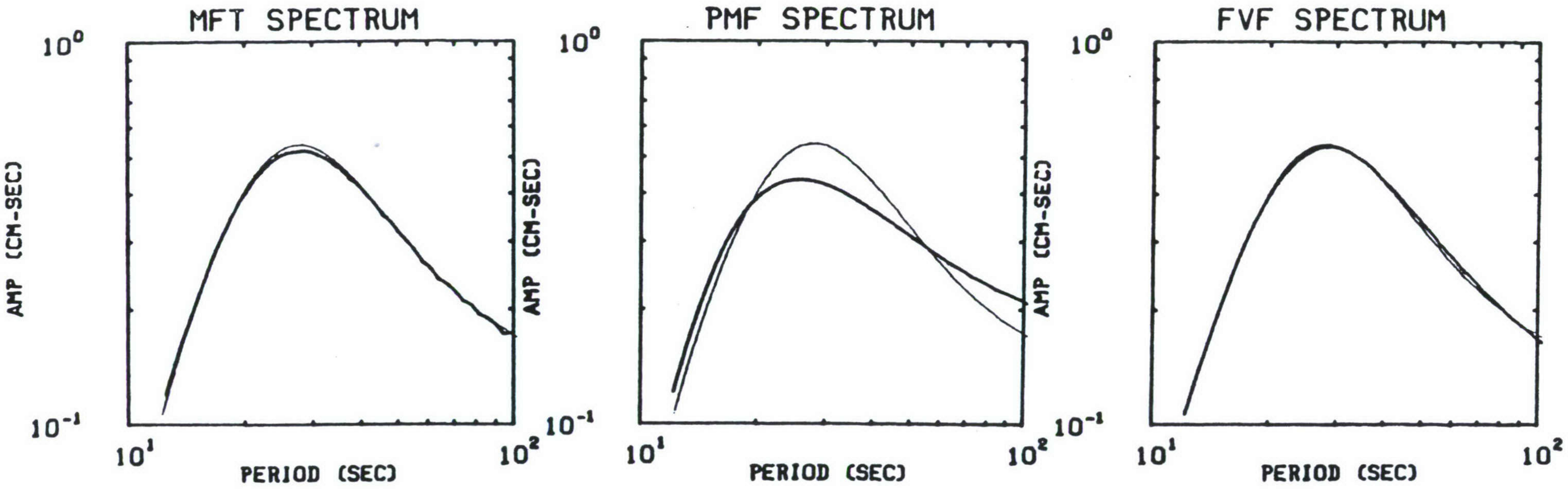

FIG. 5. Comparison of the amplitude spectra for the isolated fundamental modes. HEAVY LINE: filtered fundamental modes. LIGHT LINE: theoretical fundamental mode.

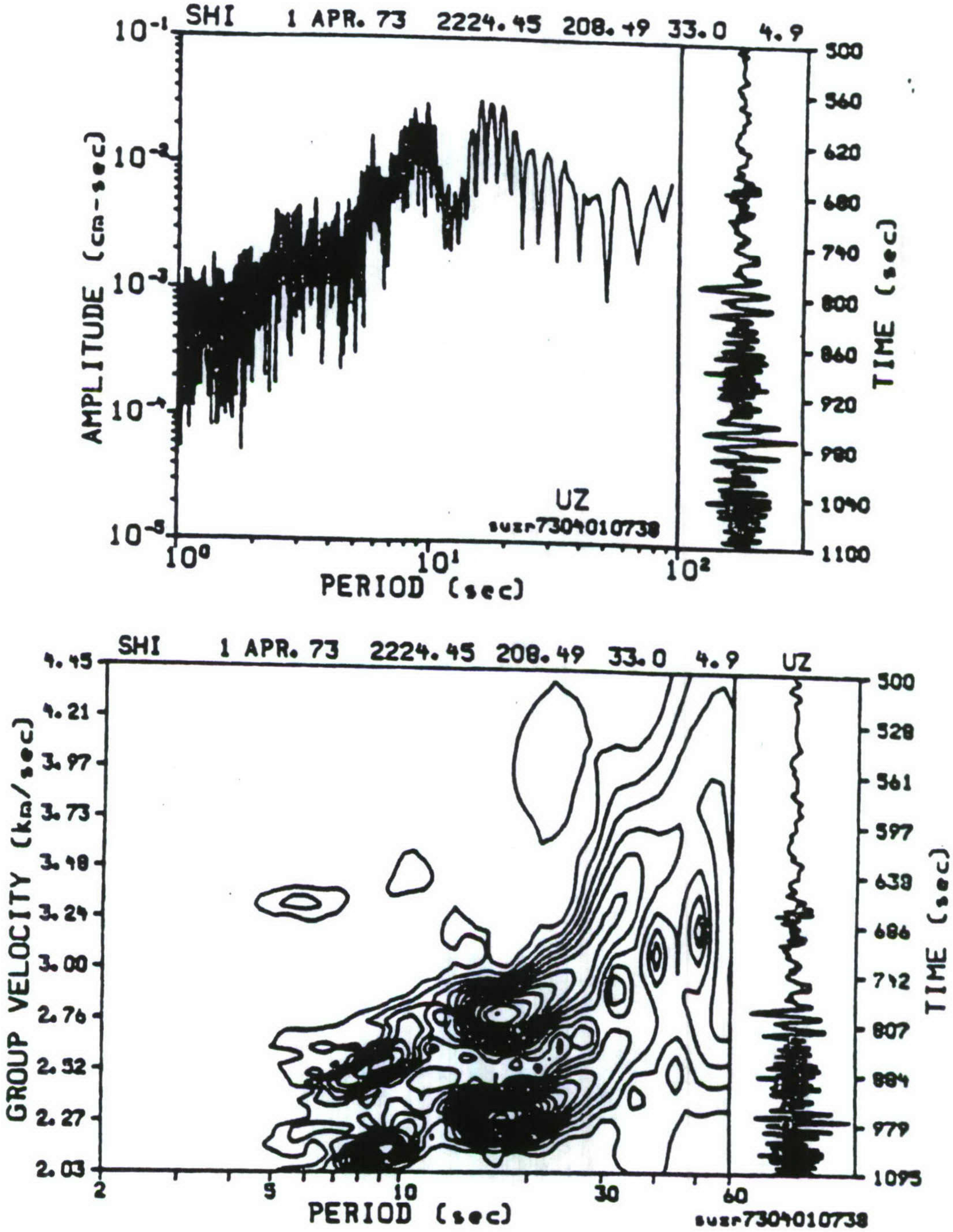

FIG. 6. Event II spectrum (top) and MFT contour plot (bottom). Time domain seismograms are plotted to the right. Notice the non-linear time scale for the MFT plot.

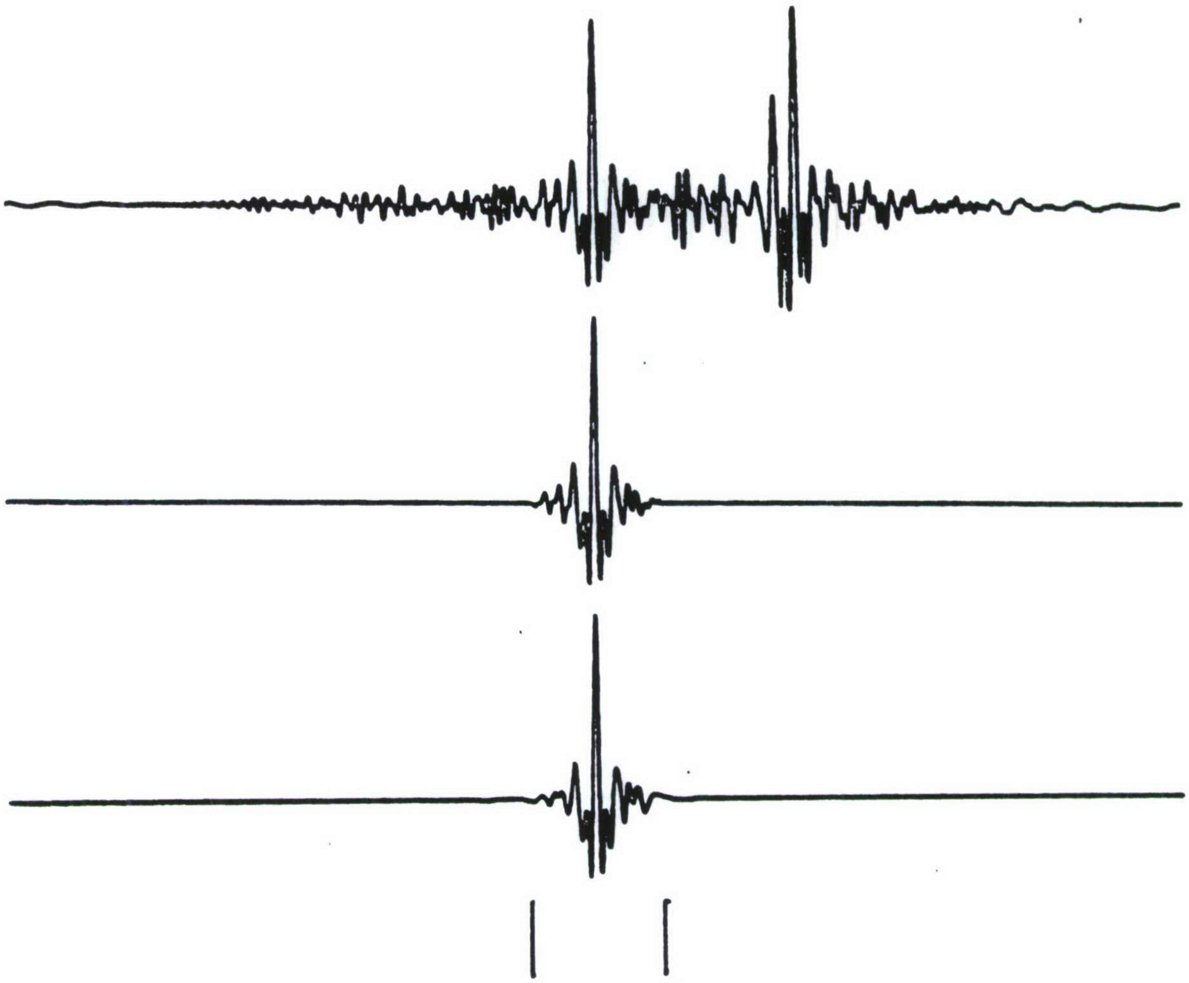

FIG. 7. Event II pseudo-autocorrelation functions. TOP: raw pseudo-autocorrelation function. MIDDLE: final pseudo-autocorrelation function for PMF. BOTTOM: final pseudo-autocorrelation function for FVF. The time marks indicate a two-sided window width of 120 seconds.

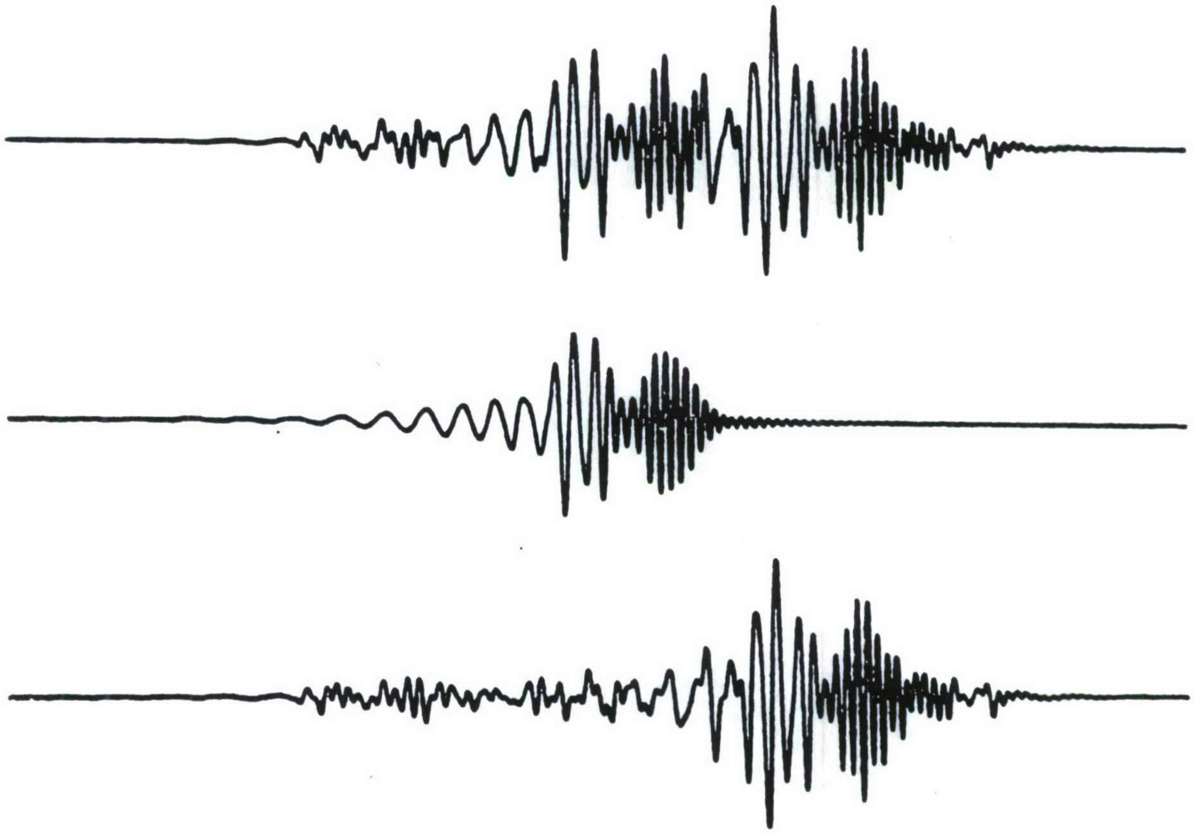

FIG. 8. Event II FVF seismograms. TOP: raw seismogram. MIDDLE: fundamental mode isolated by FVF. BOTTOM. residual obtained by subtracting the fundamental mode from the original seismogram.

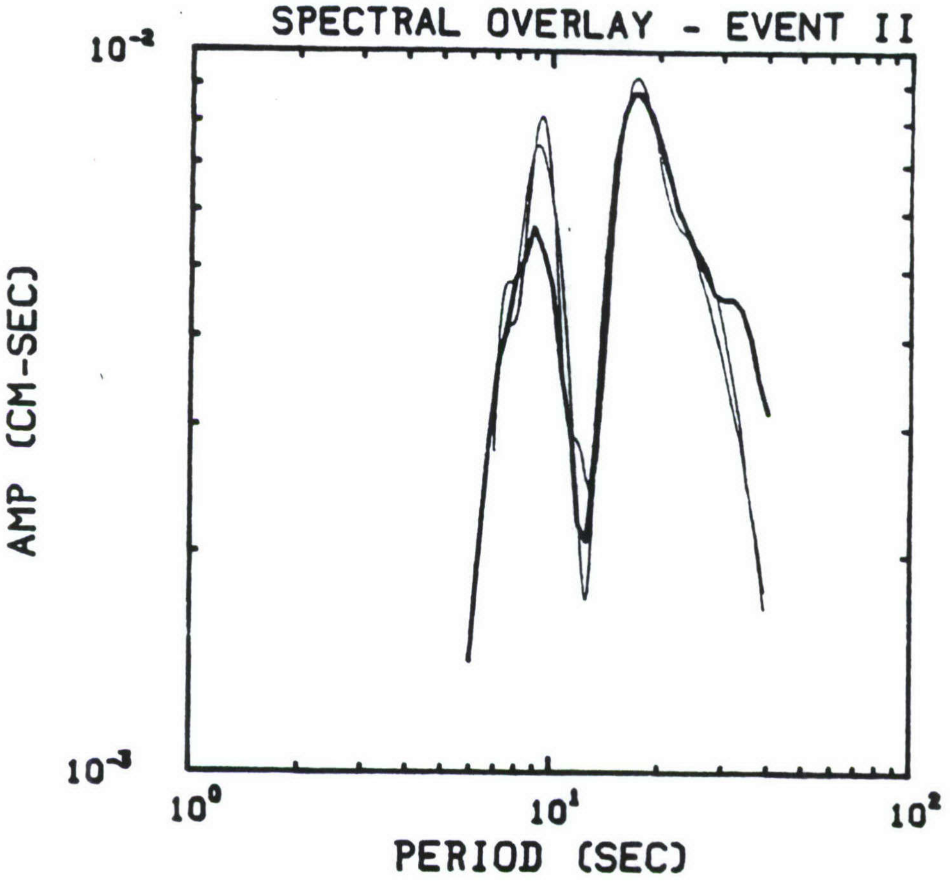

FIG. 9. Event II amplitude spectra for isolated fundamental modes. HEAVY LINE: MFT fundamental mode. LIGHT LINE: PMF and FVF fundamental modes.

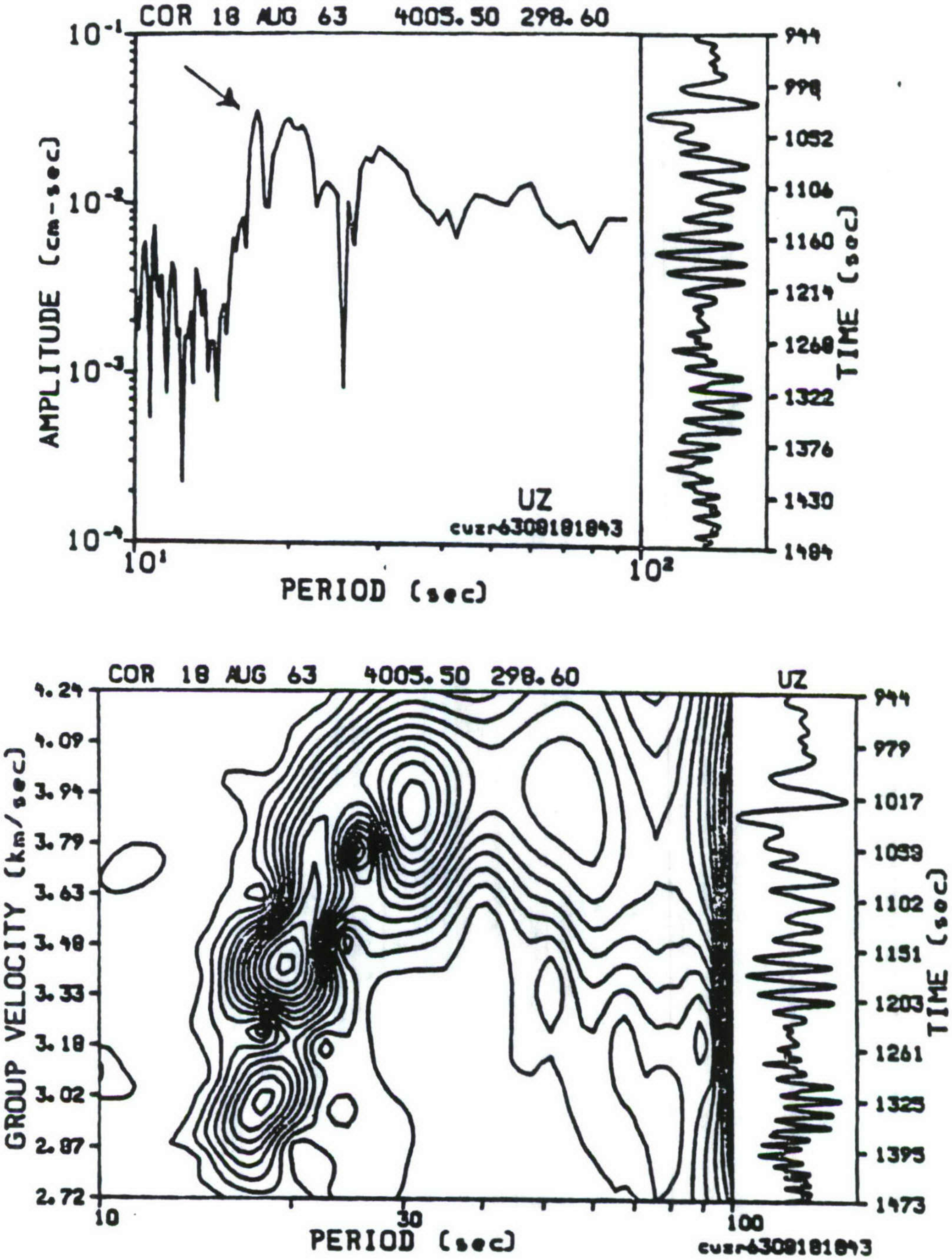

FIG. 10. Event III spectrum (top) and MFT contour plot (bottom). Time domain seismograms are plotted to the right. Notice the non-linear time scale for the MFT plot. The arrow in the upper left corner points out the location of the 17 second spectral peak referred to in the text.

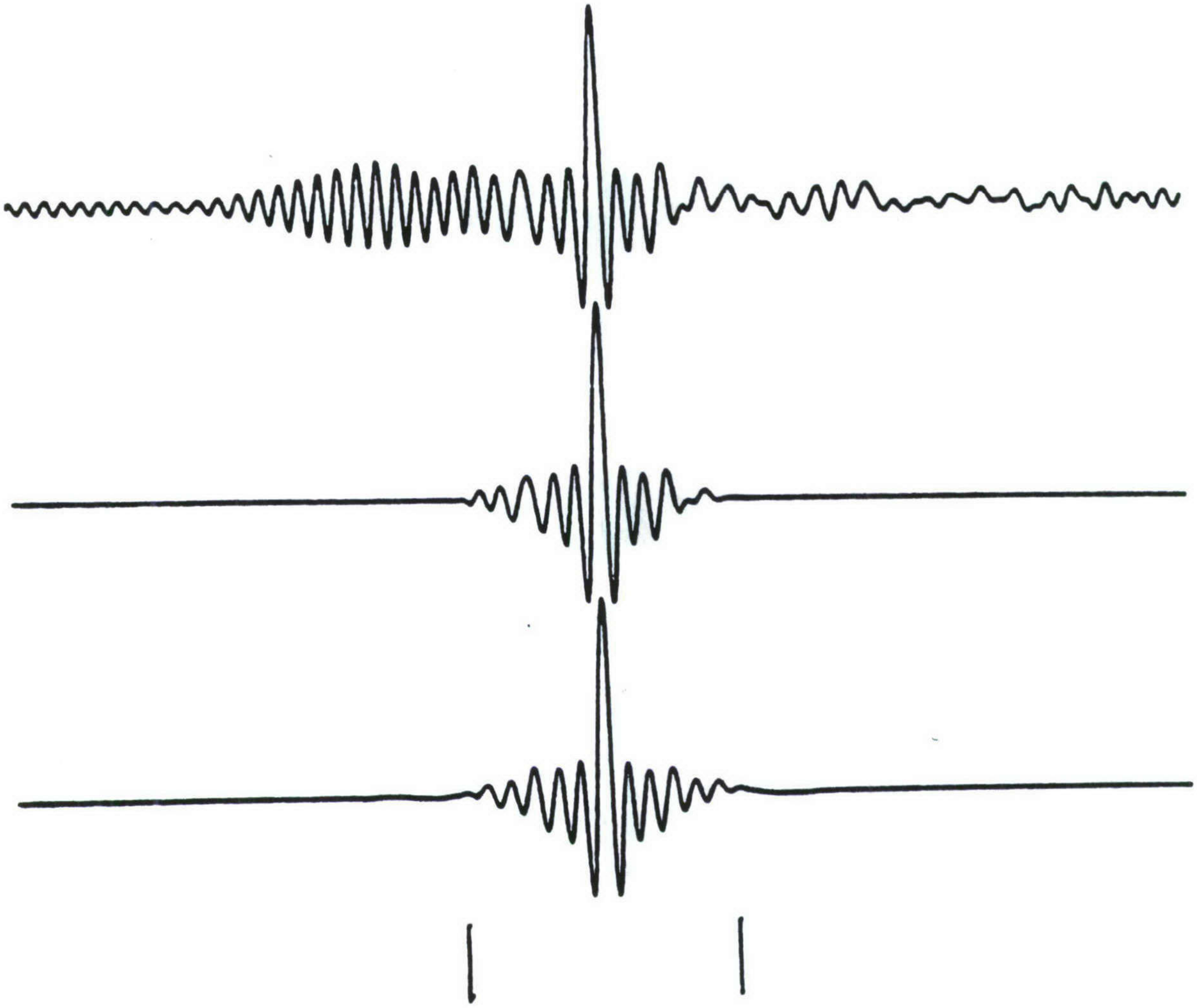

FIG. 11. Event III pseudo-autocorrelation functions. TOP: raw pseudo-autocorrelation function. MIDDLE: final pseudo-autocorrelation function for PMF. BOTTOM: final pseudo-autocorrelation function for FVF. The time marks indicate a two-sided window width of 240 seconds.

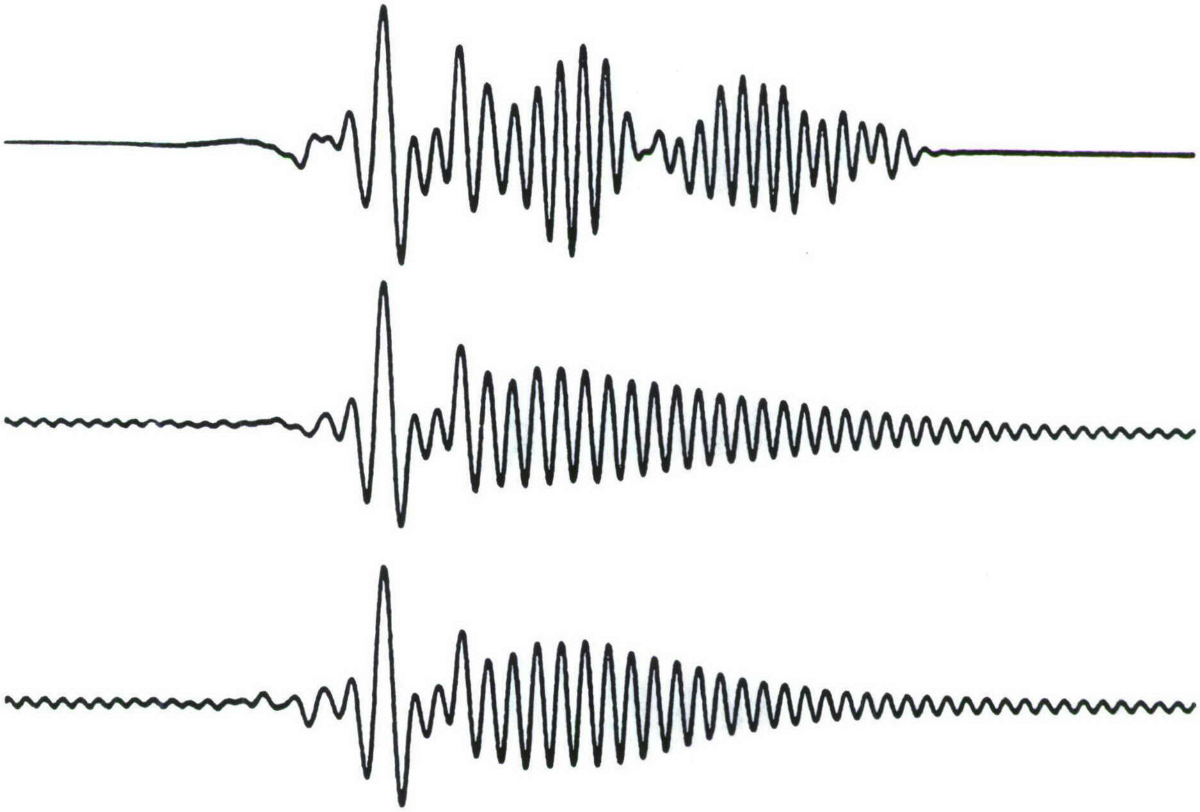

FIG. 12. Event III seismograms. TOP: raw seismogram. MIDDLE: fundamental mode isolated by PMF. BOTTOM: fundamental mode isolated by FVF.

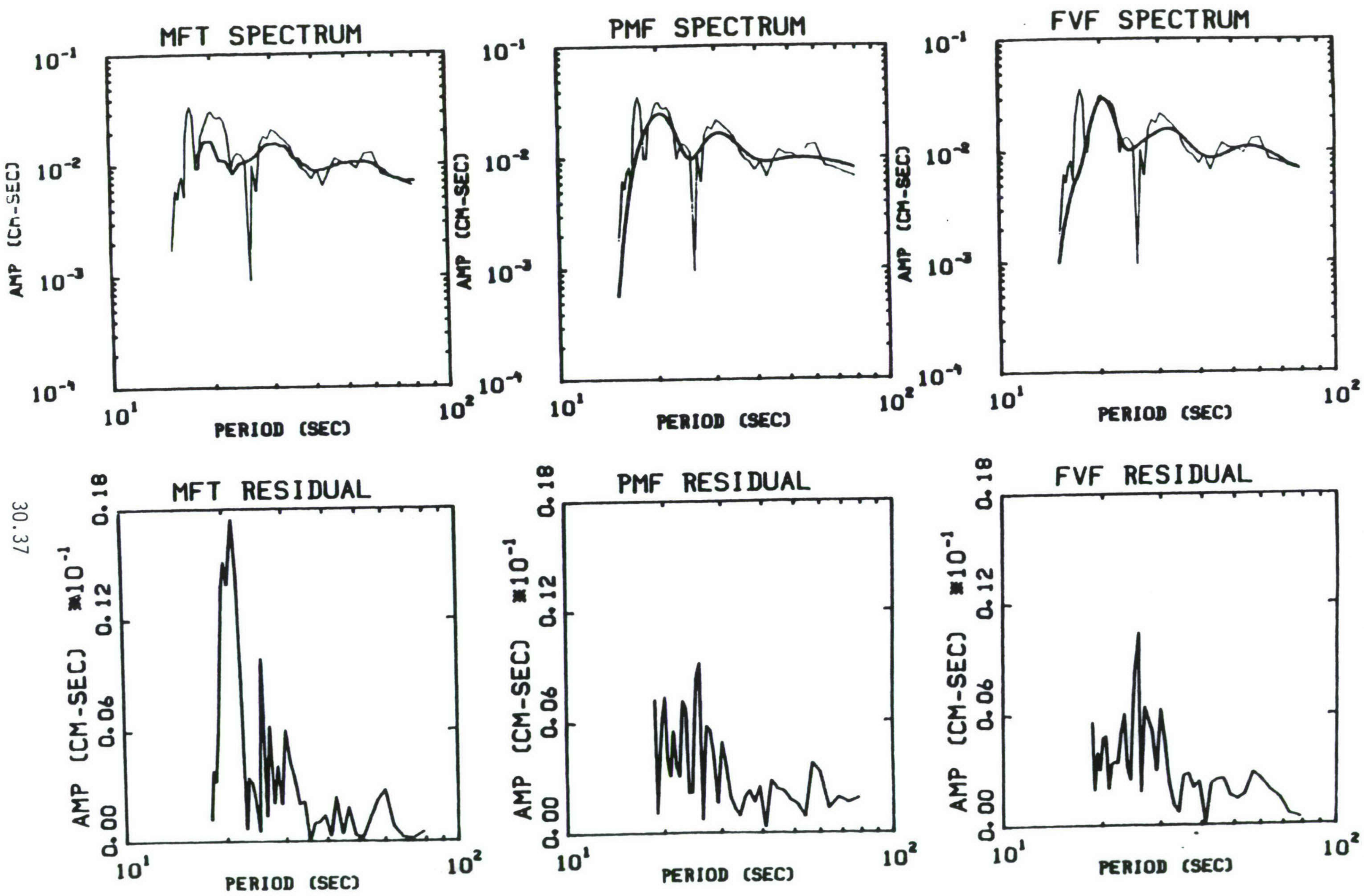

FIG 13. Event III amplitude spectra. TOP: comparison of the original amplitude spectrum with the filtered fundamental modes. LIGHT lines correspond to the original amplitude spectrum, and HEAVY lines are the filtered modes. BOTTOM: residuals between the original spectrum and the filtered modes. Notice that the vertical amplitude scale on these plots is linear.

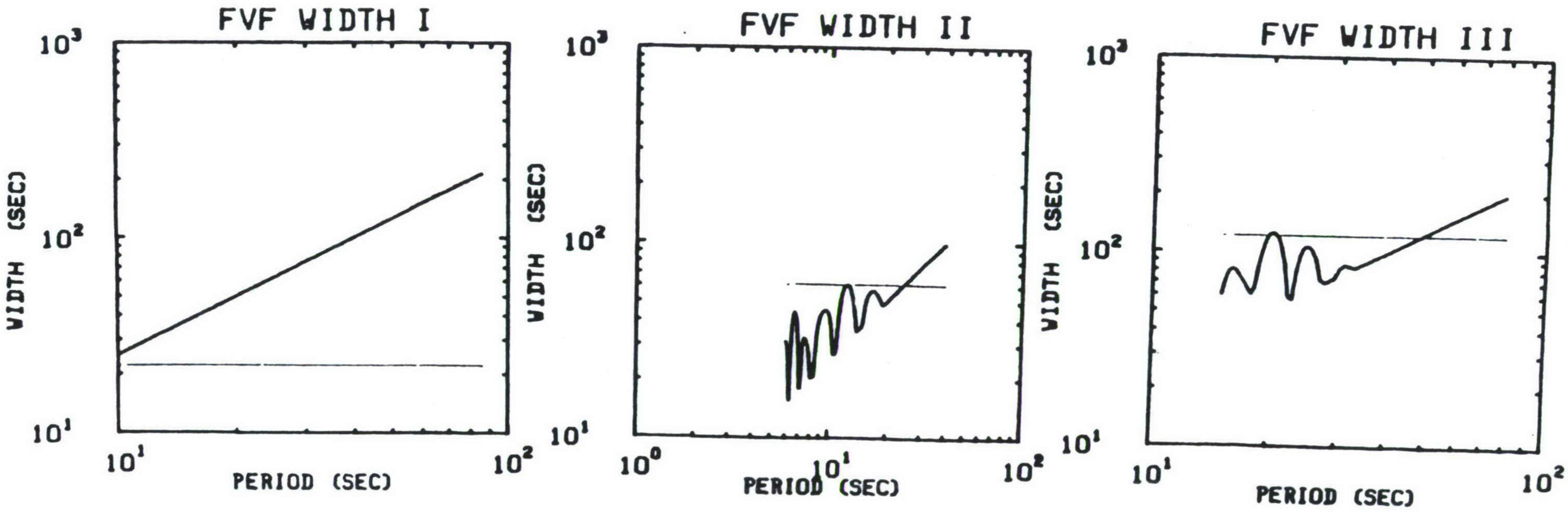

FIG. 14. FVF and PMF one-side cosine window widths calculated for events I, II, and III, respectively. HEAVY line: FVF window width. LIGHT line: PMF window width.